दिल्ली विश्वविद्यालय
University of Delhi

परिषद शाखा - II, कमरा नं: 211, नया प्रशासनिक खंड, दिल्ली विश्वविद्यालय, दिल्ली-110007, दूरभाष: 011-27001156/54
Council Branch - II, Room No. 211, New Administrative Block, University of Delhi, Delhi – 110007, Ph: 27001156/54

CNC-II/093/1/EC-1282/26/6
Dated: 29.07.2026

NOTIFICATION

Sub: Amendment to Ordinance V

(ECR 4-1-7/ dated 29.04.2026)

Following addition be made to Annexure-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

The syllabi of following Department for Semester-VII and Semester-VIII under the Faculty of Technology based on Undergraduate Curriculum Framework 2022 are notified herewith for the information of all concerned:

DEPARTMENT	SYLLABI
Computer Science and Engineering	(i) B.Tech. Computer Science and Engineering- **Annexure-1**
Electronics and Communication Engineering	(i) B.Tech. Electronics and Communication Engineering- **Annexure-2**
Electrical Engineering	(i) B.Tech. Electrical Engineering- **Annexure-3**

REGISTRAR

Annexure-1

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Detailed Course Structure and Curriculum of B.Tech. (CSE) Fourth Year

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Course Structure of B.Tech. (CSE) Fourth Year

Semester VII						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-19	Cyber Physical Systems	3	0	1	4
2	DSE-5	Select a course from the specified list of DSE-5				4
3	DSE-6 / GE-7	Select a course from the specified list of DSE-6 / GE-7				4
4	DSE-7 / GE-8	Select a course from the specified list of DSE-7 / GE-8				4
5	Dissertation on Major or Dissertation on Minor / Specialisation or Academic Project / Entrepreneurship					6
					Total Credits	**22**

Semester VIII						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-20	Fairness, Bias & Responsible Ethics in AI	3	0	1	4
2	DSE-8	Select a course from the specified list of DSE-8				4
3	DSE-9 / GE-9	Select a course from the specified list of DSE-9 / GE-9				4
4	DSE-10 / GE-10	Select a course from the specified list of DSE-10 / GE-10				4
5	Dissertation on Major or Dissertation on Minor / Specialisation or Academic Project / Entrepreneurship					6
					Total Credits	**22**

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week.
T (01 Credit) is equivalent to 01 contact hour per week.
P (01 Credit) is equivalent to 02 contact hours per week.

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department of CSE in Fourth Year

S. No.	Semester	Course Category	DSE
1.	VII	DSE-5	Web Analytics
2.			Wireless Communication
3.		DSE-6	Enterprise Application Programming
4.			Software Defined Networking
5.		DSE-7	Distributed Systems
6.			Foundations of Quantum Computing
7.	VIII	DSE-8	Bio-inspired Computing
8.			ML and DL for 3D Data
9.		DSE-9	Sustainable and Green Computing
10.			Computing for Smart Cities and Intelligent Infrastructure
11.		DSE-10	Human Centered Computing
12.			Quantum Computing and Applications

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Specializations and Minor offered by the Department of CSE

S. No.	Sem	GE	Minor in CSE (Open only for ECE/EE)	Specializations (Open only for CSE) / Minors (Open for ECE/EE)				
				Artificial Intelligence & Machine Learning	Data Science	Software Engineering	Blockchain and Cybersecurity	Augmented Reality / Virtual Reality
1	VII	DSE-6 / GE-7	Introduction to Data Analytics & Data Visualization	Edge Computing	Deep Learning Models and Applications	Software Quality Assurance and Metrics	Smart Contracts and DApps	3D Animation Design
2	VII	DSE-7 / GE-8	Fundamentals of Algorithms	Reinforcement Learning	Big Data Analytics	Software Maintenance and Evolution	Digital Currencies and Blockchain	Multiuser and Social AR/VR
3	VIII	DSE-9 / GE-9	Low level Software Systems	Generative AI	Visual Computing	Program Analysis and Verification	Cyber Law and Digital Forensics	AR/VR application development
4			OOP Concepts	Scientific Machine Learning	Applied AI	Empirical Software Engineering	Cryptanalysis & Security Models	Computer vision for AR/VR
5	VIII	DSE-10 / GE-10	Fundamentals of Machine Learning	Automated Machine Learning	Applied Generative AI for Data Science	Secure Software Engineering	AI in Cybersecurity	AR/VR Hardware Systems: Concepts and Architectures
6			Fundamentals of Cloud Computing	Federated Learning	Data Privacy, Security & Regulatory Compliance	Build, Release and Deployment Engineering	Quantum Cryptography	Ethics, Privacy, and Safety in XR Systems

Detailed Syllabus of Discipline Specific Core (DSC) Courses of B. Tech. (CSE)
SEMESTER VII

Cyber Physical Systems (DSC-19)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cyber Physical Systems	4	3	0	1	Computer Networks

Course Objectives:
1. Explain CPS concepts and characterize cyber-physical interaction using standard CPS terminology and viewpoints.
2. Develop CPS models using discrete, timed, and hybrid abstractions suitable for analysis and design.
3. Analyze timing concerns in CPS (latency, synchronization, timing-aware behavior) using NIST timing guidance and CPS timing models.
4. Design CPS data/telemetry and decision layers, including resource-aware intelligence placement.
5. Apply trustworthiness thinking (safety, security, updates, assurance) to CPS deployments.
6. Map CPS device/edge-node security requirements to the NIST IoT device cybersecurity capability baseline.

Course Outcomes:
At the end of this course, students will be able to:
1. Define CPS and justify CPS boundaries and stakeholder concerns using NIST CPS facets/aspects.
2. Construct discrete and hybrid CPS models and interpret safety requirements over system behaviors.
3. Evaluate CPS timing needs using timed models and apply scheduling/synchronization concepts to networked CPS.
4. Design CPS data/telemetry and decision layers, including resource-aware intelligence placement.
5. Design a CPS data/telemetry pipeline and assess discretization/quantization effects on decisions.
6. Position AI/ML within a CPS architecture and justify tradeoffs for reliability and resilience.
7. Produce a CPS assurance checklist: threats, secure update plan, and device capability mapping using NISTIR 8259A and CPS framework facets/aspects.

UNIT-I ***(12 hours)***
CPS Foundations and NIST CPS Framework Orientation: CPS meaning, motivation, and examples, CPS as an interaction of computational and physical dynamics, CPS modeling viewpoint (state-based and composition intuition), hybrid systems orientation, safety requirements as a CPS driver, NIST CPS Framework orientation-facets (conceptualization, realization, assurance) and aspects (functional, business, human, trustworthiness, timing, data, composition, boundaries, lifecycle), mapping a CPS scenario to facets/aspects and identifying key constraints and stakeholder needs.

UNIT-II ***(10 hours)***

CPS Modeling with Time and Real-Time Timing Foundations: Timed models for CPS, real-time scheduling fundamentals for CPS workloads, synchronization in distributed CPS and basic clock/time coordination notions, timing failures and their system impact, NIST timing themes - time awareness needs, timing/latency concerns in CPS.

UNIT-III ***(11 hours)***

CPS Data, Networks, and Decision-Making: Data vs information in CPS, information representations and limitations from data-to-model abstractions, network types and processes on networks for CPS communication and coordination, the CPS decision/action viewpoint: CPS “three-layer” framing (physical sensing/measuring layer, data/informing layer, decision/acting layer), control/decision logic for CPS: rule-based supervisors and optimization-based control; link decisions to sensing-actuation loops. Introduction to data-driven (ML-based) models in CPS as an enabling technology.

UNIT-IV ***(12 hours)***

Trustworthy CPS - Safety, Security, Assurance, and NIST-Aligned Deployment: CPS trustworthiness motivation and risk framing (cyber-physical attacks and cross-domain impacts), holistic risk management and why CPS security/safety/privacy must not be handled in silos, failures and layer-based attacks, CPS security topics, timing-security and resilient time as part of assurance, device/edge/IoT capability alignment using the NIST IR 8259A core baseline (capabilities needed from CPS-connected devices), and compiling an “assurance-ready” CPS deployment documentation set aligned to the NIST CPS facets/aspects vocabulary.

Practical Component: ***(30 hours)***

1. NIST facets/aspects mapping for a CPS scenario + constraints and stakeholders.
2. Specify inputs/outputs/state for 3 CPS components (sensor abstraction, controller abstraction, actuator abstraction) and define their interaction contract.
3. Model controller as FSM: Build a state machine for the controller logic.
4. Create a simulation of periodic sensing + control actions with variable compute/network delays; measure latency and jitter, and detect deadline misses under stress.
5. Implement a small scheduler simulation and compare missed deadlines as utilization increases.
6. Using a sensor time-series, vary sampling rate and show how discretization changes feature stability and decision confidence.
7. Compare edge vs cloud inference (latency, bandwidth, availability, privacy)
8. Implement basic data security aligned to NIST guidance
9. Runtime monitoring and audit logs, define safe-state behavior and trigger conditions
10. Integration test plan + final CPS demo + NIST-aligned system documentation (facets/aspects mapping)

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. W. M. Taha and J. Thunberg, *Cyber-Physical Systems: A Model-Based Approach*, 1st ed., Springer, 2021.
2. R. Alur, *Principles of Cyber-Physical Systems*, 1st ed., MIT Press, 2015.
3. E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed., MIT Press, 2017.

Suggested Readings:

1. R. Rajkumar, D. de Niz, and M. Klein, *Cyber-Physical Systems*, SEI Series, 1st ed., Addison-Wesley, 2017.
2. NIST CPS standards / guidance
    a. NIST SP 1500-201, Framework for Cyber-Physical Systems, Volume 1: Overview, 2017
    b. NIST SP 1500-202, Framework for Cyber-Physical Systems, Volume 2: Working Group Reports, 2017
    c. NIST SP 1500-203, Framework for Cyber-Physical Systems, Volume 3: Timing Annex, 2017
    d. NISTIR 8259A, IoT Device Cybersecurity Capability Core Baseline, 2020
    e. NIST SP 800-82 Rev.3, Guide to Operational Technology (OT) Security, 2023

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (CSE)
SEMESTER VII

WEB ANALYTICS (DSE-5)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Web Analytics	4	3	0	1	NIL

Course Objectives:

1. To introduce students to the concepts, scope, and importance of web analytics and business intelligence.
2. To understand how user interaction data is collected, measured, and analyzed from websites.
3. To familiarize students with qualitative and quantitative web analytics techniques.
4. To enable students to use web analytics tools for decision-making and performance optimization.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain fundamental concepts, terminology, and evolution of web analytics.
2. Analyze web data using qualitative and quantitative analytics methods.
3. Interpret web metrics and key performance indicators (KPIs) for optimization.
4. Apply web analytics tools such as Google Analytics for real-world website analysis.

UNIT-I ***(11 hours)***

Introduction to Web Analytics: Definition and scope of web analytics; process of web analytics; key terminology: site references, keywords and key phrases; building block terms: visit characterization, content characterization, conversion metrics; categories of web analytics: off-site web analytics and on-site web analytics; web analytics platforms; evolution of web analytics; need for web analytics; advantages and limitations of web analytics

UNIT-II ***(11 hours)***

Web Data Collection and Analysis: Data collection techniques; clickstream data: web server logs, web beacons, JavaScript tags, packet sniffing; outcomes data: e-commerce data, lead generation, brand advocacy and support; research data: mindset, organizational structure, and timing; competitive data collection: panel-based measurement, ISP-based measurement, and search engine data.

UNIT-III ***(11 hours)***

Web Metrics and Optimization: Web analytics fundamentals; capturing data using web logs and JavaScript tags; data serving and data capture; type and size of web data; integration and innovation; selecting optimal web analytics tools; clickstream data quality; identifying unique page definitions; cookies and link coding issues; Common web metrics; Introduction to KPIs.

UNIT-IV ***(12 hours)***

Advanced Web Analytics and Tools: Web analytics 2.0: limitations of web analytics 1.0 gya and evolution to analytics 2.0; competitive intelligence analysis: CI data sources, toolbar data, panel data, ISP data, search engine data, hybrid data; website traffic analysis: long-term traffic trends and competitive site analysis; Google Analytics: introduction, working, AdWords integration, benchmarking, traffic categories (organic and paid).

Practical Component: ***(30 hours)***

1. Installation and configuration of web analytics tools.
2. Collection and analysis of website traffic data.
3. Measurement of key web metrics and KPIs.
4. Website performance analysis using Google Analytics.
5. Case studies on website optimization and business intelligence.

List of Experiments:

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. B. Clifton, *Advanced Web Metrics with Google Analytics*, 2nd ed., Wiley Publishing, 2010.
2. A. Kaushik, *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity*, 1st ed., Wiley Publishing, 2009.
3. J. Sterne, *Web Metrics: Proven Methods for Measuring Web Site Success*, 1st ed., John Wiley & Sons, 2002.

Suggested Readings:

1. A. Kaushik, *Web Analytics: An Hour a Day*, 1st ed., Sybex, 2007.
2. J. Burby, S. Atchison and J. Sterne, *Actionable Web Analytics: Using Data to Make Smart Business Decisions*, 1st ed. Sybex, 2007.

WIRELESS COMMUNICATION (DSE-5)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless Communication	4	3	0	1	Computer Networks

Course Objectives:
1. To understand key principles and challenges of wireless communication systems.
2. To explore radio wave propagation effects and channel modeling.
3. To study modulation, multiple access methods, and performance analysis in wireless environments.
4. To familiarize students with modern wireless standards and system architectures.

Course Outcomes:
At the end of this course, students will be able to:
1. Explain the basic concepts and design principles of wireless communication systems.
2. Analyze wireless radio propagation and channel impairments.
3. Evaluate various modulation and multiple access schemes used in wireless systems.
4. Demonstrate understanding of cellular system architecture and modern wireless technologies.

UNIT-I ***(11 hours)***
Introduction and Cellular Fundamentals: Introduction to wireless communications; evolution of mobile communications; wireless communication systems and applications; cellular concept and frequency reuse; cell design fundamentals; structure of hexagonal cells; channel assignment and reuse pattern; handoff strategies; interference and system capacity; improving coverage and capacity.

UNIT-II ***(11 hours)***
Radio Propagation and Channel Models: Radio propagation mechanisms; free-space propagation; path loss models; large-scale vs small-scale fading; multipath propagation; Doppler effects; delay spread and coherence bandwidth; statistical channel models; shadowing and fading distributions; introduction to channel modeling.

UNIT-III ***(11 hours)***
Modulation and Multiple Access Techniques: Analog and digital modulation techniques for wireless communication: BPSK, QPSK, GMSK; spread spectrum techniques; introduction to OFDM; diversity and equalization concepts; multiple access techniques: FDMA, TDMA, CDMA, OFDMA; comparative analysis of access methods; overview of link level performance.

UNIT-IV ***(12 hours)***
Wireless Systems and Standards: Cellular system architecture; GSM system overview; introduction to 3G & 4G (WCDMA, LTE) principles; wireless LAN standards (IEEE 802.11 family); Bluetooth &

ZigBee fundamentals; introduction to 5G features; emerging paradigms (MIMO, mmWave concepts) and future directions.

Practical Component: ***(30 hours)***

1. Study of wireless system design parameters and link budget calculation.
2. Simulation of path loss and fading models.
3. Implementation of digital modulation techniques.
4. Analysis of multiple access schemes through simulation.
5. Study and configuration of wireless LANs (WiFi) in practical environments.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Pearson, 2010.
2. S. Haykin and M. Moher, *Modern Wireless Communications*, 1st ed., Pearson, 2011.
3. W. C. Y. Lee, *Mobile Communications Engineering: Theory and Applications*, 2nd ed., McGraw-Hill, 1997.

Suggested Readings:

1. A. Goldsmith, *Wireless Communications*, 1st ed., Cambridge University Press, 2012.
2. J. Schiller, *Mobile Communications*, 2nd ed., Pearson, 2008.

ENTERPRISE APPLICATION PROGRAMMING (DSE-6)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Enterprise Application Programming	4	3	0	1	Computer Networks

Course Objectives:

1. To introduce fundamentals of Enterprise Java Programming, concepts of program development using beans.
2. To familiarize students with the concept of multi threading
3. To impart knowledge about socket programming and datagrams
4. To familiarize students with the use of Servelets.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain the basic concepts and design principles of Java enterprise programming.
2. Analyze collections used for Java applications Programming as per the use case.
3. Implement multithreading and socket programming.
4. Demonstrate understanding of networking and server side, client side scripting.

UNIT-I ***(12 Hours)***

Collections**:** Collection Interfaces, Concrete Collections, Collections Framework. Multithreading : Creating and running thread, Multiple thread synchronization, Thread communication, Thread group, Thread priorities, Daemon Thread, Life Cycle of Thread.

UNIT-II ***(15 Hours)***

Fundamentals in Networking: Sockets in Java - Internet Addressing - DNS – Ipv4, IPv6 - URL class - TCP/IP and Datagram. The interfaces and classes for networking: Interfaces and classes of java.net package; InetAddress class : IP address scope - Host name resolution - Methods of InetAddress class; Program to look up the IP addresses for a hostname - Factory methods - Creating and using Sockets : Socket class - constructors and methods of Socket class. Creating TCP servers &clients: TCP/IP server sockets - Constructors and methods of ServerSocket class - Program to create a TCP/IP server and client. Handling URL: URL class - constructors and methods of URL class URLConnection class - fields of URLConnection class - methods of URLConnection class. Working with Datagrams: DatagramPacket - Constructors for DatagramPacket class - Methods of DatagramPacket class - creating Datagram server and client.

UNIT-III ***(10 Hours)***

***JDBC Package*:** JDBC – JDBC versus ODBC – Types of JDBC drivers – Connection – Statement – PreparedStatement.ResultSet :Fields of ResultSet – Methods of ResultSet – Executing a query - ResultSetMetaData – DatabaseMetaData. Datatypes in JDBC : Basic datatypes in JDBC – Advanced datatypes in JDBC – fields of Statement – methods of Statement – CallableStatement Interface – BatchUpdates

UNIT-IV ***(08 Hours)***

***Servlets* :** Using Servlets - Servlet Package - Servlet lifecycle - init() method - service() method , doGet() method, doPost() method and destroy() method . Classes and interfaces of Servlet: Servlet - GenericServlet - ServletConfig - ServletContext - ServletException - ServletInputStream - ServletOutputStream - ServletRequest – ServletResponse. Classes and interfaces of HttpServlet: HttpServlet - HttpServletRequest - HttpServletResponse - Reading HTML form data from Servlets - Response Headers - Response Redirection. Handling Servlets : Servlet Chaining - HttpUtils - Database access with JDBC inside servlet. State and Session management : Cookies - HttpSession - Server Side includes - Request forwarding – RequestDispatcher.

Practical Component: ***(30 Hours)***

1. Study of collections in Java.
2. Use of database connectivity for real time scenarios.
3. Implementation of socket programming
4. Implementation of multiple networking packages.
5. Use of multithreading for multitasking and implementation of javabeans.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. S. Holzner, *Java 2 Programming Black Book*, Dreamtech Press, 2005.
2. Herbert Schildt, Java: The Complete Reference, McGraw Hill Education, Ninth Edition
3. J. O'Neil, *JavaBeans Programming from the Ground Up*, Osborne/ McGraw-Hill TMGH, New Delhi, 1998.

Suggested Readings:

1. K. Sierra and B. Bates, *Head First EJB*, O'Reilly Media, 2003.
2. A. Goncalves, *Beginning Java EE 6 Platform with GlassFish 3: From Novice to Professional*, Apress, 2009.

SOFTWARE DEFINED NETWORKING (DSE-6)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Software Defined Networking	4	3	0	1	Computer Networks

Course Objectives:
1. To understand the fundamentals and evolution of Software Defined Networking (SDN).
2. To study the architectural principles and design of SDN systems.
3. To analyze SDN controllers, protocols, and network programmability.
4. To explore real-world deployment scenarios and performance challenges of SDN.

Course Outcomes:
At the end of this course, students will be able to:
1. Understand the need for Software Defined Networking and its advantages over traditional networks.
2. Explain SDN architecture, planes, and control mechanisms.
3. Analyze SDN protocols and controller platforms.
4. Design and evaluate SDN-based network architectures.
5. Implement basic SDN applications and network configurations.

UNIT-I ***(11 hours)***
Introduction to Software Defined Networking: Overview of traditional networking and its limitations; evolution of SDN; SDN definition and characteristics; separation of control and data planes; SDN architecture; advantages and challenges; use cases and applications of SDN.

UNIT-II ***(11 hours)***
SDN Architecture and Interfaces: Network Operating System (NOS); SDN control plane and data plane; SDN architecture models; southbound and northbound interfaces; OpenFlow architecture; flow tables; SDN communication models.

UNIT-III ***(11 hours)***
SDN Protocols and Controllers: SDN protocols and standards; OpenFlow protocol architecture; message types and flow control; software-based vs hardware-based SDN switches; SDN controllers such as POX, NOX, Floodlight, OpenDaylight; SDN controller comparison.

UNIT-IV ***(12 hours)***
SDN Design and Applications: Network programmability; Network Function Virtualization (NFV); SDN-based data centers; SDN in enterprise and cloud environments; SDN for traffic engineering; security considerations in SDN; SDN deployment challenges.

Practical Component: ***(30 hours)***

1. Study of traditional networking versus Software Defined Networking architecture.
2. Installation and configuration of SDN environment using Mininet.
3. Implementation of basic SDN topology using OpenFlow switches.
4. Configuration and testing of SDN controllers (POX / Ryu / Floodlight).
5. Flow rule creation and traffic management using SDN controllers.
6. Performance analysis of SDN-based networks under varying traffic conditions.
7. Implementation of network virtualization using SDN.
8. Case study on SDN deployment in data centers or cloud environments.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. P. Goransson, C. Black, and T. Culver, *Software Defined Networks: A Comprehensive Approach*, Morgan Kaufmann, 2016.
2. T. D. Nadeau and K. Gray, *SDN-Software Defined Networks: An Authoritative Review of Network Programmability Technologies*,O'Reilly Media, 2013.
3. W. Stallings, *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*, 1st edition, Addison-Wesley, 2015.

Suggested Readings:

1. K. Gray and T. D. Nadeau, *Network Function Virtualization*, 1st ed. Morgan Kaufmann, 2016.
2. O. Coker and S. Azodolmolky, *Software Defined Networking with OpenFlow*, Packt Publishing, 2017.

DISTRIBUTED SYSTEMS (DSE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Distributed Systems	4	3	0	1	Operating System

Course Objectives:

1. To introduce the fundamental concepts, models, and challenges of distributed systems.
2. To explain communication, coordination, and synchronization mechanisms used in distributed environments.
3. To study classical algorithms for mutual exclusion, consensus, agreement, and fault handling.
4. To expose students to modern distributed platforms, large-scale systems, and security concerns.
5. To develop the ability to analyze, compare, and reason about distributed system designs.

Course Outcomes:

After successful completion of the course, the student will be able to:

1. Explain the principles, models, and architectural foundations of distributed systems.
2. Apply communication and coordination mechanisms such as message passing, RPC, logical clocks, and snapshots in distributed environments.
3. Analyze and compare distributed algorithms for leader election, mutual exclusion, consensus, deadlock detection, and termination detection.
4. Describe the design and operation of large-scale distributed systems including P2P systems, DHTs, and distributed file systems.
5. Understand reliability, fault tolerance, and basic security mechanisms in distributed systems.

UNIT-I ***(10 hours)***

Foundations and Models: Concepts of distributed systems. Characteristics and Challenges. Case study: the world wide web. System models: Physical model, Architectural model, Fundamental models. Models of communication networks.

UNIT-II ***(12 hours)***

Communication and Coordination: Interprocess communication. Message passing systems. Remote Procedure Call (RPC). Remote Method Invocation (RMI). Sockets and message queues. Publish and Subscribe systems. Clock synchronization: Physical clocks, Logical clocks (Lamport clocks), Vector clocks. Distributed mutual exclusion, Leader election algorithms. Causal ordering of messages, Chandy-Lamport's Global State Recording Algorithm.

UNIT-III ***(12 hours)***

Mutual Exclusion, Consensus, and Agreement: Leader Election: Bully algorithm, Ring-based election. Distributed Mutual Exclusion: Requirements and challenges, Non-token-based approaches: Lamport's algorithm, Ricart-Agrawala algorithm. Quorum-based approaches, Token-based approaches: Suzuki-Kasami algorithm, Consensus and Agreement: Floodset algorithm, Byzantine General Agreement Problem. Deadlock detection. Termination Detection: Distributed termination problem, Dijkstra–Scholten algorithm.

UNIT-IV ***(11 hours)***
Modern Platforms and Case Studies: Self-Stabilization: Concept and motivation. Peer-to-Peer (P2P) Systems. Distributed Hash Tables (DHTs): Consistent hashing, Chord, Pastry. Large-Scale Distributed Systems: Google File System (GFS), Hadoop Distributed File System (HDFS), MapReduce programming model. Distributed Systems Security: Authentication, Secure communication

Practical Component: ***(30 hours)***

1. Exercises on client-server and message-passing models using socket libraries, RPC/RMI frameworks, or messaging systems.
2. Experiments on logical clocks, causal ordering, leader election, and distributed mutual exclusion using simulators or configurable distributed frameworks.
3. Hands-on exploration of consensus, deadlock detection, checkpointing, and termination detection using distributed system simulators.
4. Practical exploration of distributed file systems, MapReduce, P2P/DHTs, and security mechanisms using platforms such as Hadoop and network emulation tools.

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. Jean Dollimore, Tim Kindberg, Gordon Blair and George Coulouris.. Distributed Systems: Concepts and Design. Pearson, 2017.
2. Maarten van Steen and Andrew S. Tanenbaum. Distributed Systems. Prentice Hall of India. Third edition.
3. Sukumar Ghosh. Distributed Systems: an Algorithmic Approach. CRC Press 2015. Second edition.

Suggested Readings:

1. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly. Second edition.
2. Tom White. Hadoop - The Definitive Guide.. O'Reilly. Fourth edition.

FOUNDATIONS OF QUANTUM COMPUTING (DSE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Foundations of Quantum Computing	4	3	0	1	NIL

Course Objectives:
1. Introduce the mathematical and physical foundations of quantum computation.
2. Develop intuition about quantum phenomena such as superposition, entanglement, and measurement.
3. Enable students to understand and design quantum circuits and algorithms.
4. Provide hands-on exposure to quantum programming frameworks.
5. Bridge concepts from linear algebra, probability, and algorithms to quantum computation.
6. Prepare students for advanced study or research in quantum information science.

Course Outcomes:
At the end of this course, students will be able to:
1. Understand the fundamental principles of quantum mechanics relevant to computation.
2. Explain how quantum information differs from classical information.
3. Model quantum states, quantum gates, and quantum circuits mathematically.
4. Design and analyze basic quantum algorithms such as Deutsch–Jozsa, Grover's, and Shor's algorithms.
5. Implement simple quantum programs using quantum simulators and real quantum hardware.
6. Critically assess the potential and limitations of quantum computing for real-world problems.

UNIT-I ***(11 hours)***
Introduction to Quantum Computing:Motivation for Quantum Computing; Classical Bits vs Quantum Bits (Qubits); Historical Overview and Applications; Postulates of Quantum Mechanics (computational perspective); State Vectors and Hilbert Spaces; Dirac Notation; Measurement and Probability Amplitudes; Bloch Sphere Representation.

UNIT-II ***(11 hours)***
Mathematical Foundations:Complex Vector Spaces; Inner Products and Norms; Tensor Products and Composite Systems; Multi-Qubit States; Quantum Measurement Theory; Density Matrices (introductory); Mixed vs Pure States; Partial Measurement and Reduced Density Matrices.

UNIT-III ***(11 hours)***

Quantum Gates and Circuits:Single-Qubit Gates (Pauli, Hadamard, Phase, Rotation Gates); Multi-Qubit Gates (CNOT, Toffoli); Universal Gate Sets; Quantum Circuit Model; Circuit Depth and Complexity; Entanglement and Bell States; No-Cloning Theorem; Quantum Teleportation and Superdense Coding.

UNIT-IV ***(12 hours)***

Quantum Algorithms and Applications:Deutsch and Deutsch–Jozsa Algorithms; Grover's Search Algorithm; Shor's Factoring Algorithm (conceptual overview); Quantum Fourier Transform; Complexity Classes (P, NP, BQP – overview); Noise, Decoherence, and Error Sources; Introduction to Quantum Error Correction; Near-Term Quantum Devices and NISQ Era Applications.

Practical Component: ***(30 hours)***

1. Representation and visualization of qubits and multi-qubit states.
2. Simulation of single- and multi-qubit quantum gates.
3. Construction and simulation of quantum circuits.
4. Implementation of basic quantum algorithms using Qiskit/Cirq.
5. Running quantum programs on simulators and cloud-based quantum hardware.
6. Mini-project on a quantum algorithm or application.

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course.

Essential Readings:

1. N. D. Mermin, *Quantum Computer Science: An Introduction*, Cambridge University Press, 2007.
2. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed., Cambridge University Press, 2010.
3. J. Preskill, *Lecture Notes on Quantum Computation*, California Institute of Technology, 2015.

Suggested Readings:

1. E. G. Rieffel and W. H. Polak, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
2. Eric R. Johnston, Nic Harrigan, Mercedes Gimeno-Segovia, Programming Quantum Computers: Essential Algorithms and Code Samples, 2019.

Detailed Syllabus of Discipline Specific Core (DSC) courses for B.Tech. (CSE)
SEMESTER VIII

FAIRNESS, BIAS & RESPONSIBLE ETHICS IN AI (DSC-20)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Fairness, Bias & Responsible Ethics in AI	4	3	0	1	Artificial Intelligence & Machine Learning

Course Objectives:
1. Provide comprehensive understanding of fairness, bias, and discrimination in AI systems.
2. Examine mathematical definitions and trade-offs of fairness.
3. Introduce bias detection and mitigation techniques.
4. Study explainable, transparent, and accountable AI frameworks.
5. Explore governance, legal, and societal implications of AI deployment

Course Outcomes:
At the end of this course, students will be able to:
1. Understand foundational ethical theories and their relevance to AI systems.
2. Identify different types of bias in datasets and machine learning models.
3. Apply fairness metrics to evaluate AI systems.
4. Analyze and mitigate bias using algorithmic and data-level techniques.
5. Evaluate AI systems under responsible AI and governance frameworks.

UNIT-I ***(10 hours)***
Foundations of AI Ethics and Fairness: Introduction to AI ethics; Evolution of ethical concerns in AI; Stakeholders in AI systems; Ethical theories - Utilitarianism, Deontology, Virtue Ethics; Concepts of justice and distributive fairness; Sources of bias in AI systems - data bias, sampling bias, measurement bias, algorithmic bias, deployment bias; Individual vs group fairness; Trade-offs between fairness and accuracy.

UNIT-II ***(13 hours)***
Measuring and Mitigating Bias in AI*:* Mathematical foundations of fairness; Confusion matrix and fairness metrics; Demographic parity; Equal opportunity; Equalized odds; Predictive parity; Impossibility theorem in fairness; Bias detection through exploratory data analysis; Pre-processing methods (re-sampling, re-weighting); In-processing techniques (fair regularization, adversarial debiasing); Post-processing corrections; Fairness in classification and regression models.

UNIT-III ***(12 hours)***
Responsible, Explainable and Privacy-Preserving AI*:* Principles of Responsible AI – transparency, accountability, safety, inclusivity; Interpretability vs explainability; Model-agnostic explanation

techniques; SHAP and LIME concepts; Counterfactual explanations; Privacy concerns in AI; Differential privacy basics; Federated learning overview; Risk assessment in AI systems; Auditing AI systems; Ethical AI design frameworks and lifecycle management.

UNIT-IV ***(10 hours)***

Governance, Regulation and Societal Impact of AI: AI governance models and standards; Risk-based AI regulation; Global AI ethics guidelines; Accountability and liability in AI systems; AI in healthcare, criminal justice, finance, education, and agriculture; Human-centered AI and participatory design; Generative AI risks and misinformation; Sustainability and green AI; Case studies of AI failures and responsible redesign strategies.

Practical Component: ***(30 hours)***

1. Bias analysis on real-world datasets (e.g., hiring, loan approval).
2. Implementation of fairness metrics (demographic parity, equalized odds).
3. Comparative study of accuracy vs fairness trade-offs.
4. Application of bias mitigation techniques (pre-, in-, post-processing).
5. Explainable AI experiments using SHAP/LIME tools.
6. Mini-project: Audit and evaluate an AI system for fairness and responsibility.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. S. Barocas, M. Hardt, and A. Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*, 1st ed., MIT Press, 2023.
2. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2022.
3. L. Floridi, *Ethics of Artificial Intelligence:Principles, Challenges, and Opportunities*, 1st Ed., Oxford University Press, 2019.

Suggested Readings:

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed., MIT Press, 2016.
2. C. O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*, 1st ed., Crown Publishing, 2016.
3. IEEE, *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems*, 1st ed., IEEE, 2019.

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (CSE) SEMESTER VIII

BIO-INSPIRED COMPUTING (DSE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Bio-inspired Computing	4	3	0	1	Analysis and Design of Algorithms

Course Objectives:

1. Introduce the biological motivation behind modern computational intelligence techniques.
2. Provide a strong foundation in evolutionary and swarm-based optimization algorithms.
3. Enable students to understand and implement neuro-inspired and physics-based algorithms.
4. Develop hands-on skills in applying bio-inspired techniques to engineering and data-driven problems.
5. Encourage analytical comparison of bio-inspired approaches with classical optimization techniques.
6. Prepare students for research and advanced applications in intelligent systems and optimization.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand the fundamental principles of bio-inspired computing and how biological systems inspire computational models.
2. Analyze and apply evolutionary algorithms such as Genetic Algorithms and Differential Evolution for optimization problems.
3. Implement swarm intelligence techniques including Particle Swarm Optimization and Ant Colony Optimization for real-world problems.
4. Apply neural and neuro-inspired computing concepts for learning and pattern recognition tasks.
5. Design hybrid bio-inspired solutions combining multiple algorithms for complex optimization and decision-making problems.

UNIT-I ***(08 hours)***

Introduction to Bio-Inspired Computing: Definition and scope of bio-inspired computing; Historical evolution and motivation; Biological systems as computational models; Classification of bio-inspired algorithms: evolutionary, swarm-based, neural-inspired, and physics-based algorithms; Fitness functions and optimization landscapes; Performance measures and convergence behavior; Applications of bio-inspired computing.

UNIT-II ***(14 hours)***
***Evolutionary Algorithms**:* Introduction to evolutionary computation; Genetic Algorithms (GA): encoding schemes, population initialization, fitness evaluation, selection methods, crossover operators, mutation operators, and elitism; Genetic Programming; Evolution Strategies; Differential Evolution; Constraint handling techniques; Applications of evolutionary algorithms to function optimization, scheduling, and combinatorial problems.

UNIT-III ***(11 hours)***
***Swarm Intelligence**:* Swarm intelligence principles and collective behavior; Particle Swarm Optimization (PSO): velocity and position update equations, parameter tuning, convergence characteristics; Ant Colony Optimization (ACO): pheromone model, probabilistic path selection, pheromone update rules; Bee Colony Optimization; Applications of swarm intelligence to routing, clustering, load balancing, and optimization problems.

UNIT-IV ***(12 hours)***
Neural and Hybrid Bio-Inspired Systems: Neuro-inspired computing fundamentals; Artificial Neural Networks as bio-inspired models; Learning rules and adaptation; Hybrid bio-inspired algorithms (evolutionary-neural and swarm-neural systems); Physics- and chemistry-inspired algorithms: Simulated Annealing, Bacterial Foraging Optimization; Case studies in optimization, pattern recognition, and decision support systems; Emerging trends and research challenges in bio-inspired computing.

Practical Component: ***(30 hours)***
1. Implementation of Genetic Algorithms for optimization problems such as function minimization or Travelling Salesman Problem.
2. Implementation of Differential Evolution and comparison with Genetic Algorithms.
3. Simulation of Particle Swarm Optimization for continuous optimization problems.
4. Implementation of Ant Colony Optimization for shortest path or routing problems.
5. Design and training of a simple neural network for classification using bio-inspired learning principles.
6. Implementation of a hybrid bio-inspired algorithm combining swarm intelligence and neural networks.
7. Mini-project applying a bio-inspired algorithm to a real-world optimization problem.

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:
1. X.S. Yang, *Nature-Inspired Optimization Algorithms*, 2nd ed., Elsevier, 2021.
2. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed., Springer, 2015.
3. N. Siddique and H. Adeli, *Nature-Inspired Computing: Physics and Chemistry-Based Algorithms*, 1st ed., Chapman & Hall/CRC, 2017.

Suggested Readings:
1. Y. Xiao, *Bio-Inspired Computing and Networking*, 1st ed., CRC Press / Taylor & Francis, 2011.
2. S. S. Bhoi et al., *Bio-Inspired Neurocomputing*, 1st ed., Springer Nature, 2020.
3. X.-S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, 2nd ed., Wiley, 2010.

ML AND DL FOR 3D DATA (DSE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
ML and DL for 3D Data	4	3	0	1	Artificial Intelligence & Machine Learning

Course Objectives:

1. Provide strong probabilistic and statistical foundations for 3D data learning.
2. Introduce geometric representations and theoretical foundations of 3D shape analysis.
3. Study deep learning architectures tailored for 3D data modalities.
4. Develop practical competence in solving real-world 3D vision problems.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain mathematical foundations underlying 3D data analysis.
2. Apply probabilistic machine learning techniques to 3D shape representations.
3. Design deep neural architectures for voxel, point cloud, and mesh data.
4. Implement 3D object recognition, segmentation, and reconstruction models.
5. Evaluate and compare learning models for 3D computer vision applications.

UNIT-I ***(12 hours)***

Mathematical and Probabilistic Foundations for 3D Learning: Probability theory and distributions; Bayesian inference and decision theory; linear models for regression and classification; mixture models and EM algorithm; dimensionality reduction including PCA; graphical models; density estimation and latent variable models as discussed in Pattern Recognition and Machine Learning; geometric transformations, coordinate systems, projective geometry and camera models; image formation and multi-view geometry fundamentals as presented in Computer Vision: Models, Learning, and Inference.

UNIT-II ***(10 hours)***

Fundamentals of 3D Shape Analysis and Geometric Representations*:* 3D shape representations including point sets, surfaces, meshes and volumetric grids; differential geometry concepts; curvature and surface descriptors; shape similarity measures; feature extraction and invariant representations; registration and alignment techniques; statistical shape models; spectral methods for shape analysis; theoretical foundations and applications of 3D shape matching and retrieval as covered in 3D Shape Analysis: Fundamentals, Theory, and Applications.

UNIT-III ***(13 hours)***

Deep Learning Architectures for 3D Data*:* Introduction to deep learning for structured and unstructured 3D data; 3D convolutional neural networks for volumetric representations; multi-view CNN approaches; point-based neural networks; hierarchical feature learning for point clouds; graph neural networks for mesh processing; loss functions and optimization strategies for 3D tasks; regularization and generalization in deep models; self-supervised and transfer learning approaches for 3D data as discussed in Deep Learning for 3D Data.

UNIT-IV ***(10 hours)***

Advanced Topics in 3D Understanding and Applications: 3D object classification and segmentation; scene understanding and semantic labeling; 3D object detection and LiDAR-based perception; shape completion and reconstruction; generative models for 3D shapes; probabilistic graphical models for spatial reasoning; performance evaluation metrics for 3D tasks; integration of probabilistic learning with deep 3D architectures; applications in robotics, autonomous systems, medical imaging and augmented reality.

Practical Component: ***(30 hours)***

1. Implementation of probabilistic models(for eg PCA) for 3D feature datasets.
2. 3D data preprocessing including mesh handling, point cloud normalization and voxelization.
3. Feature-based 3D shape classification using classical machine learning methods.
4. Implementation of 3D convolutional neural networks.
5. Point cloud classification and segmentation using deep learning frameworks.
6. 3D object segmentation and reconstruction experiments.
7. Mini-project involving development of an end-to-end 3D object recognition, detection, or reconstruction system.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., Springer, 2006.
2. S. J. D. Prince, *Computer Vision: Models, Learning, and Inference*, 2nd ed., Cambridge University Press, 2012.
3. H. Shao and X. Wang, *Deep Learning for 3D Data*, 1st ed., 2023.

Suggested Readings:

1. D. W. Jacobs and R. Basri, *3D Shape Analysis: Fundamentals, Theory, and Applications*, 1st ed., 2017.
2. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2004.
3. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed., MIT Press, 2016.

SUSTAINABLE AND GREEN COMPUTING (DSE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Sustainable and Green Computing	4	3	0	1	Cloud Computing

Course Objectives:
1. To introduce the concepts and goals of sustainable and green computing.
2. To understand the environmental impact of computing systems and IT infrastructure.
3. To study energy-efficient techniques in software, hardware, data centers, and cloud computing.
4. To analyze the role of IT in achieving environmental sustainability.

Course Outcomes:
Upon completion of the course, students will be able to:
1. Analyze the environmental impact of IT and propose sustainable solutions.
2. Develop software and systems that prioritize energy efficiency and sustainability.
3. Design green data centers and cloud architectures that support sustainable practices.
4. Understand and contribute to the achievement of Sustainable Development Goals through technology.
5. Engage in responsible and ethical computing practices that promote digital sustainability.

UNIT-I ***(12 hours)***

Introduction to Green and Sustainable Computing: Introduction to Green Computing, Environmental Concerns and the importance of Sustainable Development, Environmental Impact of Information Technology, Carbon Footprint of Computing Devices, Green IT concepts: Green IT 1.0 and Green IT 2.0 (overview), Life Cycle of Computing Devices, Three R's of Sustainability: Reduce, Reuse, Recycle, Green Hardware Concepts: Energy-Efficient Devices, Reuse, Recycling, and Disposal, Green IT standards and eco-labeling, Challenges and Future Trends in Sustainable Computing.

UNIT-II ***(10 hours)***

Sustainable Software and Energy-Efficient Systems: Introduction to sustainable software development, Energy consumption in software and applications, Software sustainability attributes: efficiency, usability, maintainability, reusability, Measuring Energy Consumption in Computing Systems, Power-aware Operating Systems, Energy-Efficient Processors, Memory, and Storage, Introduction to Virtualization and its role in optimizing resources, Role of algorithms in energy-efficient computing.

UNIT-III ***(11 hours)***

Green Data Centers and Cloud Computing: Introduction to Data Centers, Energy Challenges in Modern Data Centers, Green Data Center Concepts and Design, Cooling techniques and thermal management, Green Cloud Computing and Environmental Sustainability, Features of Clouds Enabling Green Computing, Energy Efficiency of Cloud Computing, Green Cloud Architecture, Role of virtualization in green cloud computing, Benefits and challenges of green cloud computing, Enterprise Green IT strategy, Organizational considerations for Green IT Implementation.

UNIT-IV ***(12 hours)***

Sustainable Development Goals, Socio-Ethical Aspects: Introduction to Sustainability and the United Nations Sustainable Development Goals (SDGs), Role of Green and Sustainable Computing in achieving SDGs, Sustainable Cities and Communities: Smart Cities and Green ICT solutions, Intelligent Transportation Systems and Energy-aware Urban Systems, Sustainable Hardware Lifecycle, E-waste Management and Circular Economy in IT, Socio-cultural impact of Green IT, Responsible user practices and digital sustainability, Ethics, privacy, and security in sustainable information systems, Green IT governance, policies, and standards, Emerging technologies for sustainable computing.

Practical Component: ***(30 hours)***

1. To study the concept of Green Computing and analyze the environmental impact of computing systems.
2. To study the Three R's of Sustainability (Reduce, Reuse, Recycle) in IT systems.
3. To analyze power management features in operating systems.
4. To evaluate the energy efficiency of different algorithms.
5. To study the concept of Virtualization for resource optimization.
6. To analyze the energy consumption in data centers.
7. To study the architecture of Green Cloud Computing.
8. To study the role of green computing in sustainable urban systems.
9. To design a Green IT strategy for an organization.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. S. Murugesan and G. R. Gangadharan, *Harnessing Green IT: Principles and Practices*, 1st ed., Wiley-IEEE Computer Society, 2012.
2. J. Lamb, *The Greening of IT*, 1st ed., IBM Press, 2009.
3. J. Harris, *Green Computing and Green IT: Best Practices on Regulations and Industry*, 1st ed., Emero, 2008.
4. W. Leonhard and K. Murray, *Green Home Computing for Dummies*, 1st ed., For Dummies, 2009.

Suggested Readings:

1. S. K. Das, *Cloud Computing*, 1st ed., Dominant Pub and Dist, 2015.
2. P. Pattnaik and M. Kabat, *Fundamentals of Cloud Computing*, 1st ed., Vikas Publishing House, 2014.

COMPUTING FOR SMART CITIES AND INTELLIGENT INFRASTRUCTURE (DSE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Computing for Smart Cities and Intelligent Infrastructure	4	3	0	1	NIL

Course Hours:L-03, T-00, P-02

Course Objectives:
1. To understand the multi-layered architecture of Smart City solutions.
2. To explore the role of edge and cloud computing in managing urban data.
3. To analyze intelligent infrastructure systems like smart grids, ITS, and waste management.
4. To address the security, privacy, and sustainability challenges in hyper-connected cities.

Course Outcomes:
By the end of this course, the student will be able to:
1. Design a communication framework for large-scale sensor networks in urban settings.
2. Apply machine learning algorithms to predict urban trends (traffic, energy demand).
3. Evaluate the efficiency of intelligent transport and energy systems.
4. Develop prototypes for smart urban services using IoT and Cloud platforms.

UNIT-I ***(10 hours)***
Smart City Framework and Connectivity: The Smart City Ecosystem: Definition, dimensions and the 4-layer architecture. Communication Protocols: Short-range vs. Long-range. 5G and Beyond: The role of ultra-low latency in vehicle-to-everything (V2X) communication. Urban Sensing: Mobile crowdsensing, fixed sensor networks, and satellite imagery integration.

UNIT-II ***(12 hours)***
Intelligent Infrastructure Systems: Smart Energy & Grids: Demand-response management, integration of renewable energy, and smart metering. Intelligent Transportation Systems: Real-time traffic monitoring, adaptive signal control, and autonomous shuttle integration. Smart Water & Waste: Automated leak detection, quality monitoring, and sensor-based waste collection routing. Structural Health Monitoring: Using IoT to monitor the integrity of bridges, buildings, and tunnels.

UNIT-III ***(13 hours)***
Data Analytics and Urban Intelligence: Edge vs. Cloud Computing: Distributed processing for real-time urban decision-making. Urban Big Data: Challenges of volume, velocity, and variety in city-scale data. Predictive Modeling: Time-series analysis for air quality forecasting and energy consumption prediction. Digital Twins: Creating virtual replicas of physical city infrastructure for simulation and stress testing.

UNIT-IV ***(10 hours)***

Governance, Security, and Sustainability: Smart Governance: Open data portals, e-participation, and blockchain for transparent city records. Cybersecurity in Infrastructure: Protecting critical assets from hacking. Anonymization techniques for citizen data. Circular economy models and the role of computing in reducing the carbon footprint of cities.

Practical Component : ***(30 hours)***

1. Sensor Integration: Interface an air quality sensor (MQ series) with an ESP32 and push data to a cloud dashboard.
2. Traffic Simulation: Use SUMO to model traffic flow and test adaptive signal timing.
3. Develop a Python script to predict household energy consumption using a public Smart Meter dataset and LSTM networks.
4. Design a system using ultrasonic sensors and an MQTT broker to display real-time parking availability.
5. Configure a LoRa gateway and node to measure signal strength (RSSI) across different urban obstacles.
6. Use a platform to visualize real-time sensor data overlaid on a 3D building model.
7. Implement a simplified smart contract for a community-based "Participatory Budgeting" system.

List of Experiments

Note: Course instructor may add/delete/update new experiments in addition to the above Suggested practical exercise

Essential Readings:

1. Smart Cities: Foundations, Principles, and Applications — by H. Song, R. Srinivasan, T. Sookoor, S. Jeschke
2. Internet of Things: A Hands-On Approach — by A. Bahga & V. Madisetti
3. Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia — by Anthony M. Townsend

Suggested Readings:

1. Shalli, et al. (2022). IoT and WSN based Smart Cities: A Machine Learning Perspective. Springer.
2. Pardo, T. A., & Nam, T. (2011). Conceptualizing smart city with dimensions of technology, people, and institutions.

HUMAN CENTERED COMPUTING (DSE-10)

CREDIT DISTRIBUTION, ELIGIBILITY, AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Human Centered Computing	4	3	0	1	NIL

Course Objectives:

1. Develop a strong foundation in Human-Centered Design (HCD) for building usable, useful, and inclusive computing systems.
2. Train students to perform user research, translate insights into requirements, and iteratively design through prototyping.
3. Build practical capability in interaction design for modern computing contexts (web/mobile, data-intensive products, AI-assisted interfaces).
4. Introduce accessibility, human factors, ethics, and privacy so students can design responsibly for diverse users and real-world constraints.
5. Enable students to evaluate and improve designs using standard Human Computer Interaction (HCI) / (User Experience) UX evaluation methods, metrics, and evidence-based iteration.

Course Outcomes:

At the end of this course, students will be able to:

1. Plan and conduct user research and synthesize findings into personas, scenarios, task flows, and measurable usability goals.
2. Produce interaction designs and prototypes (low and high fidelity) for web/mobile and other interactive computing experiences.
3. Apply accessibility and inclusive design principles and identify/remediate common usability barriers.
4. Design user experiences for data-driven and AI-assisted features with transparency, user control, and calibrated trust.
5. Conduct usability evaluations (expert and user-based), analyze results, and propose prioritized design improvements.
6. Incorporate privacy-by-design and ethical considerations into interface requirements and product decisions.

UNIT-I ***(10 hours)***

Human Factors as System Constraints and Interaction as Computation: Human-Centered Computing (HCC) scope as socio-technical systems engineering, cognition and attention limits, mental models and conceptual models, human error types and recovery design, responsiveness and perceived performance, interaction as event-driven computation, modeling interaction with state machines and invariants, multimodal inputs (touch/voice/gesture) and sensor/actuator feedback loops in interactive devices, accessibility engineering fundamentals (semantic UI, keyboard-first flows, screen-reader compatibility), inclusive design constraints for diverse abilities and contexts, ethics primer for user impact and manipulation resistance (dark patterns, consent-as-interaction).

UNIT-II ***(15 hours)***

Engineering Interaction Architecture for Modern Computing Systems: User-work modeling as engineering artifacts (personas, hierarchical task inventory, task sequences, flow models), information architecture as graphs (navigation, findability, search), interaction architecture from task-flow → screen-flow → API contract, component-based UI architecture, design systems and reusable components (tokens, theming, consistency), client-server interaction design (API affordances, pagination/infinite scroll, optimistic UI, offline-first, caching), performance engineering (render cost, network waterfalls, critical path, caching/CDN considerations), collaboration and social computing primitives (shared state, concurrency, consistency, presence/awareness cues, moderation primitives), ubiquitous/IoT and XR interaction considerations (context, safety boundaries, sensor uncertainty, ergonomic constraints), domain-driven HCC examples (health, education, finance tech) - constraints, compliance, and real-world workflows.

UNIT-III ***(10 hours)***

Data-Driven HCC-Instrumentation, Experiments, Visualization, and Decision Support: Telemetry and event schema design, product analytics and behavioral measurement limits, metrics taxonomy (UX quality metrics, guardrails, business metrics) and target setting, funnel/cohort/retention analysis, causal reasoning intuition for product decisions, online controlled experiments (A/B tests), experimentation platform architecture (feature flags, staged rollouts, canaries), observability for human outcomes (error budgets for UX, rage clicks, abandonment, crash loops), dashboards and visualization principles for trustworthy decision support (cognitive load, uncertainty display, misleading charts).

UNIT-IV ***(10 hours)***

Trustworthy Human-Centered Computing-Usable Security, Privacy, Fairness, Human-AI Interaction, and Governance: Usable security engineering (auth flows, permissions, least-privilege UX, phishing-resistant and abuse-resistant design), safety and misuse-case analysis (harassment, fraud, coercion, unsafe automation), privacy engineering in systems and products (consent, notice design, minimization, retention, sensitive data handling), trustworthy AI characteristics and risk framing for AI-enabled features (valid/reliable, safe, secure/resilient, accountable/transparent, explainable / interpretable, privacy-enhanced, fair), human-AI interaction patterns (set expectations, feedback, control, graceful failure), explanation UX and uncertainty communication by user role, feedback loops and human-in-the-loop operations (labeling UX, escalation), recommender/ranking interaction risks (manipulation, filter bubbles, over-personalization), fairness concepts and tradeoffs.

Practical Component: ***(30 hours)***

1. Choose a computing product problem (web/mobile/data/AI-based), define target users, context-of-use, and measurable success criteria
2. Conduct user research (minimum: 1 interview/observation + 1 survey), synthesize findings into key insights and pain points
3. Create personas, scenarios, user journeys, and task flows, write clear UX requirements and acceptance criteria
4. Build low-fidelity prototypes (paper/wireframes), run quick feedback sessions, iterate
5. Build a clickable/high-fidelity prototype (Figma or a simple front-end), ensure consistency using a mini design system
6. Define error states and recovery flows, design micro-interactions and feedback for key tasks
7. Accessibility audit and improvement plan (contrast, typography, keyboard navigation, focus order, labels), implement fixes in the prototype

8. Design a data-centric screen (dashboard/analytics/search) with usable filters, progressive disclosure, and clear hierarchy
9. Design an AI-assisted feature (recommendation/autocomplete/copilot-style help) with transparency, user control, and fallback behavior
10. Implement a human-in-the-loop flow for the AI feature (confirm/override/feedback capture), define the feedback data structure and logging
11. Run a usability test (5-8 users), compute SUS + task metrics, analyze results and prioritize fixes
12. Add privacy UX elements (consent, settings, data controls, retention notice), justify choices using NIST Privacy Framework thinking, final demo + report

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human-Computer Interaction*, 6th ed., Wiley, 2023.
2. R. Hartson and P. S. Pyla, *The UX Book: Agile UX Design for a Quality User Experience*, 3rd ed., Morgan Kaufmann / Elsevier, 2025.
3. W3C, *Web Content Accessibility Guidelines (WCAG) 2.2 and ARIA Authoring Practices Guide (APG)*, World Wide Web Consortium, latest edition.
4. R. Kohavi, D. Tang, and Y. Xu, *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*, Cambridge University Press, 2020.

Suggested Readings:

1. Microsoft, *Human-AI Interaction Guidelines*, Microsoft Research, latest edition.
2. Google PAIR Team, *People + AI Guidebook and PAIR Design Patterns*, Google Research, latest edition.
3. A. Mathur, M. Kshirsagar, and J. Mayer, “Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites,” *Proceedings of the ACM on Human-Computer Interaction*, ACM, 2019.
4. NIST standards / guidance
    a. NIST, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, National Institute of Standards and Technology, 2023.
    b. NIST, *NIST Privacy Framework: A Tool for Improving Privacy through Enterprise Risk Management*, Version 1.0, National Institute of Standards and Technology, 2020.

QUANTUM COMPUTING AND APPLICATIONS (DSE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Quantum Computing and Applications	4	3	0	1	Foundations of Quantum Computing

Course Objectives:

1. Build advanced theoretical and practical skills in quantum computing.
2. Introduce cutting-edge quantum algorithms and application domains.
3. Enable students to work with noisy intermediate-scale quantum (NISQ) devices.
4. Provide hands-on experience with quantum programming and optimization.
5. Encourage interdisciplinary applications across AI, cryptography, and physical sciences.
6. Prepare students for research or industry roles in quantum technologies.

Course Outcomes:
At the end of this course, students will be able to:

1. Analyze and design advanced quantum algorithms for computational problems.
2. Apply quantum machine learning techniques to data-driven tasks.
3. Understand quantum error correction and fault-tolerant computation.
4. Evaluate quantum hardware architectures and noise models.
5. Implement hybrid quantum–classical algorithms on real quantum devices.

UNIT-I ***(11 hours)***
Advanced Quantum Algorithms*:* Amplitude Amplification and Variants; Advanced Applications of Grover's Algorithm; Quantum Phase Estimation; Quantum Walks; Variational Quantum Algorithms (VQE, QAOA); Hamiltonian Simulation; Linear Systems Algorithms (HHL – conceptual understanding).

UNIT-II ***(11 hours)***
Quantum Error Correction and Noise*:* Sources of Noise and Decoherence; Noise Models (Depolarizing, Dephasing, Amplitude Damping); Quantum Error Correction Codes (Bit-flip, Phase-flip, Shor Code, Steane Code); Fault-Tolerant Quantum Computation; Threshold Theorem; Error Mitigation Techniques for NISQ Devices.

UNIT-III ***(11 hours)***
Quantum Machine Learning and Optimization*:* Quantum Data Encoding Techniques; Quantum Kernels; Variational Quantum Classifiers; Quantum Neural Networks; Hybrid Quantum–Classical Learning; Quantum Optimization Algorithms; Applications in Pattern Recognition, Chemistry, and Finance.

UNIT-IV ***(12 hours)***

Applications and Emerging Directions*:* Quantum Cryptography and Post-Quantum Security; Quantum Simulation in Chemistry and Materials Science; Quantum Computing for Optimization and Logistics; Quantum Sensing and Metrology (overview); Cloud-Based Quantum Computing Platforms; Ethical, Societal, and Security Implications of Quantum Technologies.

Practical Component: ***(30 hours)***

1. Implementation of variational quantum algorithms (VQE, QAOA).
2. Simulation and mitigation of noise in quantum circuits.
3. Quantum machine learning models using hybrid frameworks.
4. Execution of quantum programs on real quantum hardware.
5. Performance benchmarking of quantum vs classical algorithms.
6. Capstone mini-project focused on a real-world quantum application.

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course.

Essential Readings:

1. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed., Cambridge University Press, 2010.
2. M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, 1st ed., Springer, 2018.
3. J. Preskill, *Quantum Computing in the NISQ Era and Beyond*, Quantum, vol. 2, 2018.

Suggested Readings:

1. A. Montanaro, *Quantum Algorithms: An Overview*, npj Quantum Information, vol. 2, 2016.
2. Jack D. Hidary, *Quantum Computing: An Applied Approach*, Springer, 2nd Edition, 2019.

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors / Specializations by the Department of CSE in Fourth Year

1. Minor in CSE (Offered to ECE and EE)

a. GE-7: Introduction to Data Analytics & Data Visualization
b. GE-8: Fundamentals of Algorithms
c. GE-9: Low Level Software Systems
d. GE-9: OOP Concepts
e. GE-10: Fundamentals of Machine Learning
f. GE-10: Fundamentals of Cloud Computing

2. Minor/Specialization in Artificial Intelligence & Machine Learning (Offered to CSE, ECE, and EE)

a. DSE-6 / GE-7: Edge Computing
b. DSE-7 / GE-8: Reinforcement Learning
c. DSE-9 / GE-9: Generative AI
d. DSE-9 / GE-9: Scientific Machine Learning
e. DSE-10 / GE-10: Automated Machine Learning
f. DSE-10 / GE-10: Federated Learning

3. Minor/Specialization in Data Science (Offered to CSE, ECE, and EE)

a. DSE-6 / GE-7: Deep Learning Models and Applications
b. DSE-7 / GE-8: Big Data Analytics
c. DSE-9 / GE-9: Visual Computing
d. DSE-9 / GE-9: Applied AI
e. DSE-10 / GE-10: Applied Generative AI for Data Science
f. DSE-10 / GE-10: Data Privacy, Security & Regulatory Compliance

4. Minor/Specialization in Software Engineering (Offered to CSE, ECE, and EE)

a. DSE-6 / GE-7: Software Quality Assurance and Metrics
b. DSE-7 / GE-8: Software Maintenance and Evolution
c. DSE-9 / GE-9: Program Analysis and Verification
d. DSE-9 / GE-9: Empirical Software Engineering
e. DSE-10 / GE-10: Secure Software Engineering
f. DSE-10 / GE-10: Build, Release and Deployment Engineering

5. Minor/Specialization in Blockchain and Cybersecurity (Offered to CSE, ECE, and EE)

a. DSE-6 / GE-7: Smart Contract and DApps

b. DSE-7 / GE-8: Digital Currencies and Blockchain
c. DSE-9 / GE-9: Cyber Law and Digital Forensics
d. DSE-9 / GE-9: Cryptanalysis & Security Models
e. DSE-10 / GE-10: AI in Cybersecurity
f. DSE-10 / GE-10: Quantum Cryptography

6. Minor/Specialization in Augmented Reality / Virtual Reality (Offered to CSE, ECE, and EE)

a. DSE-6 / GE-7: 3D Animation Design
b. DSE-7 / GE-8: Multiuser and Social AR/VR
c. DSE-9 / GE-9: AR/VR Application Development
d. DSE-9 / GE-9: Computer Vision for AR/VR
e. DSE-10 / GE-10: AR/VR Hardware Systems: Concepts and Architectures
f. DSE-10 / GE-10: Ethics, Privacy, and Safety in XR Systems

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of CSE in SEMESTER VII

INTRODUCTION TO DATA ANALYTICS & DATA VISUALIZATION (GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Data Analytics & Data Visualization	4	3	0	1	NIL

Course Objectives:

1. Introduce the principles and workflow of data analytics.
2. Develop statistical thinking for analyzing engineering data.
3. Teach data cleaning, transformation, and exploratory analysis techniques.
4. Provide practical skills in data visualization tools and libraries.
5. Demonstrate applications of analytics in electrical and electronics engineering domains.

Course Outcomes:

Upon completion of the course, students will be able to:

1. Understand fundamental concepts of data analytics and its role in engineering applications.
2. Apply statistical and exploratory techniques to analyze engineering datasets.
3. Perform data preprocessing and transformation for real-world data.
4. Create effective data visualizations for technical interpretation and decision-making.
5. Analyze domain-specific datasets from ECE and EE applications.

UNIT-I ***(09 hours)***

Foundations of Data Analytics: Introduction to data analytics and its importance in engineering; types of data including structured and unstructured data; data analytics lifecycle; descriptive, diagnostic, predictive and prescriptive analytics; data sources in real world such as sensor data, power system measurements and communication signals; basic probability concepts; measures of central tendency and dispersion; introduction to data-driven decision making.

UNIT-II ***(14 hours)***

Data Preprocessing and Exploratory Data Analysis: Data collection and data formats; handling missing data and outliers; data cleaning and normalization; data transformation and feature engineering; correlation analysis; covariance and regression basics; exploratory data analysis techniques; visualization for exploration; time-series data analysis for power and signal data; introduction to simple statistical modeling.

UNIT-III ***(10 hours)***

Data Visualization Techniques: Principles of effective visualization; types of plots including line plots, bar charts, histograms, scatter plots and box plots; visualization of multivariate data; time-series visualization; heatmaps and correlation matrices; dashboards and reporting; storytelling with data; visualization best practices for engineering reports; introduction to visualization tools.

UNIT-IV ***(12 hours)***

Analytics Applications in Real World: Signal data analysis; power system load analysis; fault detection using data analytics; predictive maintenance basics; basic machine learning concepts for engineers including supervised and unsupervised learning; clustering and classification overview; case studies involving communication networks, smart grids and IoT systems; ethical considerations in data analytics.

Practical Component: ***(30 hours)***

1. Data collection and preprocessing using Python or equivalent tools.
2. Implementation of statistical measures and exploratory data analysis.
3. Visualization of engineering datasets using plotting libraries.
4. Time-series analysis of power or sensor data.
5. Mini-project involving analysis and visualization of a real-world dataset.

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course.

Essential Readings:

1. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.
2. D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, 7th ed., Wiley, 2018.
3. W. McKinney, *Python for Data Analysis*, 3rd ed., O'Reilly Media, 2022.

Suggested Readings:

1. C. N. Knaflic, *Storytelling with Data: A Data Visualization Guide for Business Professionals*, 1st ed., Wiley, 2015.
2. F. Provost and T. Fawcett, *Data Science for Business*, 1st ed., O'Reilly Media, 2013.

EDGE COMPUTING (DSE-6 / GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Edge Computing	4	3	0	1	NIL

Course Objectives:

1. Introduce edge computing as a distributed computing paradigm spanning the cloud–edge continuum, driven by latency, bandwidth, privacy, and reliability needs.
2. Build skills to design edge system architectures, choose appropriate platforms (containers, orchestration, MEC), and reason about deployment constraints.
3. Develop competency in edge resource management (placement, offloading, caching, scheduling) and performance evaluation.
4. Enable students to build data/streaming and AI-at-edge solutions, including model deployment and monitoring under edge constraints.
5. Introduce security, privacy, and operational concerns for edge deployments, including standards-based edge / MEC architectural thinking.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain edge computing concepts, motivations, and architectures, and differentiate edge, fog, cloudlets, and MEC-style deployments.
2. Design an end-to-end edge solution including edge nodes, connectivity, data pipeline, and cloud integration, and justify architectural choices using performance and locality constraints.
3. Implement and deploy edge services using containers and basic orchestration patterns, and evaluate latency/throughput/resource usage.
4. Build an edge analytics/AI workflow (stream ingestion to inference to feedback/monitoring) and apply suitable optimization patterns for edge constraints.
5. Identify major edge security/privacy risks and apply baseline mitigation strategies and operational practices in an edge prototype.

UNIT-I ***(10 hours)***

Edge Computing Fundamentals and the Cloud–Edge Continuum: edge computing motivations (low latency, bandwidth reduction, data locality, resilience, privacy), edge vs cloud vs fog vs cloudlets vs MEC, edge deployment scenarios (industrial IoT, smart cities, AR/VR, video analytics, connected vehicles) and workload characteristics, edge reference architectures (device-gateway-edge node–regional cloud-central cloud), data locality and context awareness, service models (edge as platform, edge as cache, edge analytics), edge application decomposition into microservices, introduction to edge performance metrics (end-to-end latency, tail latency, jitter, availability, energy), overview of canonical edge vision and drivers from foundational literature.

UNIT-II ***(12 hours)***

Edge Platforms, Virtualization, and Orchestration: edge hardware and system constraints (heterogeneity, intermittent connectivity, limited power), virtualization choices (VMs vs containers, lightweight isolation), container-based deployment and packaging, edge orchestration concepts (service discovery, placement, scaling, rolling updates), state management at the edge (stateless vs stateful services, local storage, synchronization), networking for edge (service meshes at a high level, NAT constraints, secure tunnels), mobility and session continuity concepts, MEC framework and reference architecture notions (MEC system and host, platform services, reference points), integration of edge with 4G/5G ecosystems and SDN/NFV ideas at a conceptual level.

UNIT-III ***(12 hours)***

Edge Resource Management, Offloading, and Data Pipelines: computation offloading motivations and models (partial vs full offload, device–edge–cloud split), placement and scheduling basics (latency-aware placement, resource-aware scheduling, multi-tenancy), edge caching and content distribution concepts, stream processing at the edge (windowing, event-time vs processing-time intuition, backpressure basics), data management and consistency tradeoffs (eventual consistency, synchronization, conflict handling), QoS/QoE thinking (SLA-like targets, admission control), benchmarking and evaluation methodology (workload profiling, measurement pitfalls, tail latency focus), overview of research directions and system challenges highlighted in surveys.

UNIT-IV ***(11 hours)***

Edge Intelligence, Security, Privacy, and Operational Reliability: edge analytics and edge AI (on-device/near-device inference, pipeline design for sensor/video/text workloads), model deployment patterns (batch vs streaming inference, model versioning at a high level), optimization themes (model compression/quantization concepts, hardware acceleration awareness, latency budgeting), federated/distributed learning intuition and privacy-aware learning concepts, trust and transparency in edge AI decisions, security threat landscape (insecure edge nodes, rogue gateways, data leakage, model theft/poisoning at a high level), security controls (identity, authentication, secure communication, isolation, secure updates, monitoring and logging), reliability patterns (graceful degradation, offline operation, failover, observability), reading and discussing modern edge+AI research directions from recent survey/tutorial literature.

Practical Component: ***(30 Hours)***

1. Edge lab setup: configure an edge node (Raspberry Pi / x86 mini-PC / VM), basic Linux hardening, networking, time sync, benchmarking tools
2. Containerization: package a microservice (REST/gRPC) into Docker, run locally, measure latency and resource usage
3. Edge messaging pipeline: MQTT/Kafka-lite/NATS-style pub-sub (any one), sensor/event simulation, logging and replay
4. Edge gateway pattern: device → gateway → edge node, local buffering during connectivity loss, replay on reconnection
5. Simple MEC-style decomposition exercise: split an application into edge vs cloud components, justify split using latency/bandwidth constraints
6. Orchestration mini-lab: deploy services on a lightweight edge orchestrator (e.g., k3s or docker-compose), rolling update and basic service discovery
7. Offloading experiment: implement and compare local vs edge vs cloud execution for a CPU-bound task, measure end-to-end and tail latency
8. Edge caching experiment: cache hot content/results at edge, compare hit rate, latency, bandwidth savings
9. Edge stream analytics: implement windowed aggregation/anomaly rule over streaming data, handle backpressure via batching/throttling

10. Edge AI inference: deploy a lightweight model for classification/anomaly detection, measure inference latency and accuracy tradeoff
11. Edge AI robustness: simulate drift/noise and log model confidence/false alarms, propose mitigation (thresholding, retraining trigger, fallback)
12. Final mini-project: build an end-to-end edge application (video analytics / industrial telemetry / smart campus / health monitoring simulator), include deployment scripts, evaluation report, and a short discussion of security and reliability controls.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. R. Buyya and S. N. Srirama (eds.), *Fog and Edge Computing: Principles and Paradigms*, 1st ed., Wiley, 2019.
2. J. Taheri and S. Deng (eds.), *Edge Computing: Models, Technologies and Applications*, 1st ed., Institution of Engineering and Technology (IET), 2020.
3. L. Xu and W. Shi (eds.), *Edge Computing: Systems and Applications*, 1st ed., Wiley / Wiley-IEEE Press, 2025.

Suggested Readings:

1. ETSI, *Multi-access Edge Computing (MEC): Framework and Reference Architecture (ETSI GS MEC 03)*, latest ed., European Telecommunications Standards Institute (ETSI).
2. M. Satyanarayanan, “The Emergence of Edge Computing,” *Computer*, IEEE Computer Society, vol. 50, no. 1, pp. 30–39, 2017.
3. W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge Computing: A Survey,” *Future Generation Computer Systems*, vol. 97, pp. 219–235, Elsevier, 2019.
4. H. Hua et al., “Edge Computing with Artificial Intelligence: A Machine Learning Perspective,” ACM Computing Surveys, vol. 55, no. 9, ACM, 2023.

DEEP LEARNING MODELS AND APPLICATIONS (DSE-6 / GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Deep Learning Models and Applications	4	3	0	1	Fundamentals of Data Analytics, Artificial Intelligence and Machine Learning

Course Objectives:

1. Introduce foundations and mathematical principles of deep learning.
2. Understand and implement major deep learning architectures.
3. Apply deep learning models to vision, language, and sequence problems.
4. Analyze real-world applications and challenges of deep learning systems.

Course Outcomes:

Upon completion of the course, students will be able to:

1. Explain core concepts and learning mechanisms of deep neural networks.
2. Design and implement deep learning models using standard architectures.
3. Apply deep learning techniques to real-world vision and language problems.
4. Evaluate performance and address challenges in deep learning applications

UNIT-I ***(12 hours)***

Foundations of Deep Learning: Introduction to Deep Learning, ML vs DL, Biological inspiration, Perceptron, MLP, Activation functions, Loss functions, Gradient descent, Backpropagation, Regularization and optimization challenges.

UNIT-II ***(12 hours)***

Deep Learning Architectures: CNN architecture, convolution, pooling, padding, stride, CNN variants, RNN, vanishing gradient, LSTM, GRU, Attention mechanism, Transformer overview.

UNIT-III ***(12 hours)***

Deep Learning Models for Vision and Language: Computer vision tasks, AlexNet, VGG, ResNet (overview), NLP with deep learning, word embeddings, sequence modeling, Vision Transformers, transfer learning.

UNIT-IV ***(09 hours)***

Applications and Advanced Topics in Deep Learning: Applications in speech, vision, healthcare, recommender systems, autonomous systems, Autoencoders, VAEs, GANs, evaluation metrics, ethical issues, case studies.

Practical Component: ***(30 Hours)***

1. Implementation of basic neural networks using Python
2. CNN-based image classification

3. Sequence modeling using RNN / LSTM
4. Transfer learning using pre-trained models
5. Mini-project on deep learning application

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course.

Essential Readings:

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed., MIT Press, 2016.
2. F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2021.
3. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O'Reilly Media, 2022.

Suggested Readings:

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., Springer, 2006.
2. S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., Pearson Education, 2009.

SOFTWARE QUALITY ASSURANCE AND METRICS (DSE-6 / GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Software Quality Assurance and Metrics	4	3	1	0	Software Engineering, Object Oriented Software Engineering

Course Objectives:

1. To introduce concepts of software quality and quality assurance.
2. To develop a quantitative understanding of software measurement and metrics.
3. To apply product, process, and maintenance metrics for quality evaluation.
4. To analyze object-oriented software quality using standard metrics.
5. To expose students to software quality estimation tools and computer-aided quality engineering.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain software quality assurance principles and standards.
2. Apply software quality metrics for product, process, and maintenance evaluation.
3. Analyze object-oriented software quality using standard OO metrics.
4. Estimate and analyze software quality using quantitative models and tools.
5. Apply metrics methodology and CAQE concepts in software quality evaluation

UNIT-I ***(10 hours)***

Software Quality Concepts and Quality Assurance: Introduction to software quality; definitions and perspectives of quality; software quality attributes; software quality assurance (SQA): objectives, activities, and role in software development; verification and validation; quality planning; quality assurance vs quality control; introduction to software quality standards (ISO 9001, ISO/IEC 25010).

UNIT-II ***(12 hours)***

Software Quality Metrics and Measurement: Software measurement concepts; metrics and measurement theory; software quality metrics; product quality metrics; process quality metrics; size metrics (LOC, Function Point Analysis); complexity metrics (Cyclomatic Complexity); productivity metrics; metrics for software maintenance; limitations and misuse of metrics.

UNIT-III ***(13 hours)***

Object-Oriented Metrics and Quality Estimation: Object-oriented software engineering principles; need for object-oriented metrics; Chidamber and Kemerer (CK) metrics suite; MOOD metrics; coupling, cohesion, inheritance, and polymorphism metrics; software quality metrics methodology; Goal–Question–Metric (GQM) approach; defect density and reliability metrics; software quality estimation models.

UNIT-IV ***(10 hours)***

Software Quality Tools and Computer-Aided Quality Engineering: Software quality estimation tools: objectives and scope; desirable features of software quality estimation tools; static and dynamic analysis tools; defect tracking and metrics tools; comparative study of existing tools; Computer-Aided Quality Engineering (CAQE) concepts; design techniques for CAQE; integration of CAQE with the software development life cycle; case studies.

Tutorial Component: ***(15 hours)***

1. Identification and analysis of software quality attributes for a given system.
2. Numerical problems on LOC, Function Point Analysis, and Cyclomatic Complexity.
3. Computation and interpretation of product and process quality metrics.
4. Analysis of maintenance metrics using defect and effort data.
5. Evaluation of object-oriented designs using CK and MOOD metrics.
6. Designing a metrics plan using the Goal–Question–Metric (GQM) approach.
7. Case study on software quality estimation using metrics and models.
8. Comparative study of software quality estimation tools.
9. Conceptual design of a Computer-Aided Quality Engineering (CAQE) framework.

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Essential Readings:

1. R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill Education, 2019.
2. Y. Singh and R. Malhotra, *Object-Oriented Software Engineering*, 1st Edition, PHI Learning Private Limited, 2010.
3. N. E. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*, 3rd Edition, CRC Press, 2014.
4. S. H. Kan, *Metrics and Models in Software Quality Engineering*, 2nd Edition, Addison-Wesley, 2002.

Suggested Readings:

1. I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2015.
2. ISO/IEC 25010, *Systems and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE)*, 2011.
3. ISO 9001, *Quality Management Systems – Requirements*, 2015.

SMART CONTRACTS AND DAPPS (DSE-6 / GE-7)

CREDIT DISTRIBUTION, ELIGIBILITY, AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Smart Contracts and DApps	4	3	0	1	Blockchain Essentials

Course Objectives:

1. To understand the fundamentals of blockchain and decentralized systems.
2. To introduce the concepts of smart contracts, decentralized applications, and blockchain platforms.
3. To develop the ability to design and deploy smart contracts using Solidity.
4. To understand the architecture, tools, and security aspects of decentralized applications.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain the working principles of blockchain and Ethereum platforms.
2. Develop smart contracts using Solidity programming language.
3. Deploy and interact with smart contracts in a blockchain environment.
4. Design and implement decentralized applications using modern blockchain tools.
5. Analyze security issues and best practices in smart contract development.

UNIT-I ***(11 hours)***

Introduction to Blockchain and Ethereum: Introduction to Blockchain Technology; Evolution of Distributed Ledger Systems; Centralized vs Decentralized Architectures; Blockchain Components; Cryptographic Foundations; Ethereum Overview; Ethereum Architecture; Ethereum Virtual Machine (EVM); Accounts, Transactions, and Gas; Use Cases of Blockchain and Ethereum.

UNIT-II ***(11 hours)***

Smart Contracts and Ethereum Environment: Smart Contracts: Definition and Characteristics; Smart Contract Lifecycle; Account Types and Transactions; Gas and Gas Optimization; Ethereum Development Environment; Tools and Frameworks; Web3 and Ethereum APIs; Writing, Compiling, Testing, and Deploying Smart Contracts.

UNIT-III ***(11 hours)***

Solidity Programming: Introduction to Solidity Language; Structure of a Solidity Program; Data Types and Variables; Control Structures; Functions and Modifiers; Visibility Specifiers; Events and Logging; Error Handling (require, assert, revert); Memory, Storage, and Calldata; Contract Creation and Interaction.

UNIT-IV ***(12 hours)***

Decentralized Applications (DApps): Decentralized Application Architecture; DApp Components; Front-end and Back-end Integration; Smart Contract Interaction using Web3.js and Ethers.js; Wallet Integration (MetaMask); DApp Design Patterns; Use Cases such as Voting, Token Systems, and Auctions; Security Considerations and Best Practices.

Practical Component: ***(30 hours)***

1. Installation and configuration of blockchain development environment (Ganache, Remix, MetaMask).
2. Creation and management of cryptocurrency wallets and accounts.
3. Writing, compiling, and deploying smart contracts using Solidity.
4. Implementing token standards (ERC-20 / ERC-721) and executing transactions.
5. Developing and testing decentralized applications (DApps) using Web3.js.
6. Demonstrating smart contract security features and common vulnerabilities.
7. Case studies on real-world blockchain applications and deployments.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. A. M. Antonopoulos and G. Wood, *Mastering Ethereum*, 2nd ed., O'Reilly Media, 2018.
2. G. Shrivastava, D.-N. Le, and K. Sharma, *Cryptocurrencies and Blockchain Technology Applications*, 1st ed., Wiley & Sons, 2020.
3. Josh Thompson, *Blockchain: The Blockchain for Beginners - Guide to Blockchain Technology and Programming*, 1st ed., Independently Published, 2017.

Suggested Readings:

1. K. Solorio, R. Kanna, and D. H. Hoover, *Hands-On Smart Contract Development with Solidity and Ethereum: From Fundamentals to Deployment*, O'Reilly Media, 2019.
2. A. M. Antonopoulos and D. A. Harding, *Mastering Bitcoin: Programming the Open Blockchain*, 3rd ed. O'Reilly Media, 2023.

3D ANIMATION DESIGN (DSE-6 / GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
3D Animation Design	4	3	0	1	NIL

Course Objectives:

1. To introduce the standard 3D production pipeline (Pre-production, Production, Post-production).
2. To master the technical aspects of polygonal modeling, UV mapping, and digital sculpting.
3. To apply the "12 Principles of Animation" within a digital 3D environment.
4. To understand the physics of lighting, shading, and rendering engines.

Course Outcomes:

By the end of this course, the student will be able to:

1. Construct complex 3D assets using industry-standard modeling techniques.
2. Apply realistic textures and materials using UV unwrapping and PBR workflows.
3. Animate characters and objects with believable physics and personality.
4. Render high-quality sequences using global illumination and ray-tracing techniques.

UNIT-I ***(15 hours)***

Modeling and Geometry: Introduction to 3D Interface: Navigating the viewport, coordinate systems (Local vs. World), and basic primitives.;Polygonal Modeling: Extrude, Bevel, Bridge, and Boolean operations; maintaining "All-Quads" topology for animation.;NURBS vs. Polygons: Understanding mathematical surfaces vs. mesh-based surfaces; Digital Sculpting: High-poly detailing and the concept of multiresolution modeling.

UNIT-II ***(11 hours)***

Surfacing and Texturing: UV Unwrapping: Projecting 3D surfaces onto 2D planes; managing texture seams and stretching; Shading Networks: Introduction to PBR (Physically Based Rendering) materials; Albedo, Roughness, and Normal maps; Procedural Texturing: Creating materials using nodes and noise functions without external images; Baking Maps: Transferring high-poly detail to low-poly models via Normal and Ambient Occlusion maps.

UNIT-III ***(11 hours)***

Rigging and Animation: Character Rigging: Skeleton creation (Joints/Bones), Forward Kinematics (FK), and Inverse Kinematics (IK); Skinning: Weight painting to define how the mesh deforms during movement; The 12 Principles: Implementing Squash and Stretch, Anticipation, Staging, and Follow-through in 3D; Graph Editor: Fine-tuning animation curves, interpolation types (Linear vs. Bézier), and keyframe timing.

UNIT-IV ***(8 hours)***

Lighting, Dynamics, and Rendering: Lighting Techniques: Three-point lighting (Key, Fill, Rim), HDRI environments, and Global Illumination.;Particle Systems: Simulating fire, smoke, and basic fluid dynamics; Rendering Engines: Differences between Rasterizers (Real-time) and Path Tracers (Arnold/Cycles); Compositing: Basic multi-pass rendering (Render Layers) and final video output.

Practical Component: ***(30 hours)***

List of Experiments:

1. Hard Surface Modeling: Design and model a detailed mechanical object (e.g., a vintage camera or a sci-fi drone).
2. Character Sculpting: Create a basic humanoid or creature bust using digital sculpting tools.
3. UV & Texturing Lab: Perform a "clean" UV unwrap of a complex object and apply hand-painted or PBR textures.
4. The Bouncing Ball: Animate a ball with different weights (Bowling ball vs. Ping-pong ball) to master Squash and Stretch.
5. Character Walk Cycle: Create a seamless 24-frame looping walk cycle for a bipedal character.
6. Interior Lighting: Setup an architectural visualization scene and render it using Physical Sky and Area lights.

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. R. Williams, *The Animator's Survival Kit*, Faber & Faber, 2001.
2. D. Derakhshani, *Introducing Autodesk Maya*, Sybex, 2014.
3. A. Chopine, *3D Art Essentials: The Fundamentals of 3D Modeling, Texturing, and Animation*, Focal Press, 2011.

Suggested Readings:

1. W. Vaughan, *Digital Modeling*, New Riders, 2011.
2. A. Beane *3D Animation Essentials: New Directions for Evaluation,,* John Wiley, 2012.

FUNDAMENTALS OF ALGORITHMS (GE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Fundamentals of Algorithms	4	3	0	1	Data Structures

Course Objectives:

1. Introduce core principles of algorithms in an application-oriented manner.
2. Develop the ability to estimate computational efficiency using asymptotic notations.
3. Familiarize students with commonly used algorithms in engineering analysis.
4. Enable practical problem solving using structured algorithmic techniques.
5. Strengthen logical reasoning and analytical thinking skills across disciplines.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand basic concepts of algorithms and their role in engineering problem solving.
2. Analyze algorithm efficiency using simple time and space complexity measures.
3. Apply searching and sorting techniques to engineering datasets.
4. Use fundamental algorithmic approaches to solve optimization and graph-based engineering problems.

UNIT-I ***(10 hours)***

Introduction to Algorithms and Complexity: Definition and characteristics of algorithms; algorithm design steps; flowcharts and pseudocode; performance analysis; time and space complexity; best case, worst case and average case analysis; asymptotic notations including Big O, Big Theta and Big Omega; practical examples from signal processing, power systems and numerical computation.

UNIT-II ***(11 hours)***

Searching, Sorting and Recursion: Linear search and binary search; introduction to sorting concepts; bubble sort, selection sort and insertion sort; overview of merge sort and quick sort (conceptual understanding); comparison of sorting algorithms; basic idea of recursion and recursive algorithms; simple applications in engineering data processing; complexity comparison of searching and sorting techniques.

UNIT-III ***(12 hours)***

Algorithmic Strategies for Problem Solving*:* Introduction to divide and conquer with practical examples; greedy method fundamentals; applications such as activity selection and minimum spanning tree (conceptual); introduction to dynamic programming with simple examples such as Fibonacci and shortest path in grid; comparison of greedy and dynamic programming approaches; engineering optimization problems modeled algorithmically.

UNIT-IV ***(11 hours)***

Graphs and Engineering Applications: Introduction to graphs and their representation; traversal techniques including BFS and DFS; shortest path algorithms including Dijkstra's algorithm; minimum spanning trees and their applications in network design; topological sorting (conceptual); applications of graph algorithms in communication networks, power grids, transportation and circuit analysis; basic introduction to algorithmic problem modeling in engineering systems.

Practical Component: ***(30 hours)***

1. Implementation of searching and sorting algorithms.
2. Complexity comparison using runtime measurement.
3. BFS and DFS implementation on engineering network graphs.
4. Dijkstra's algorithm for shortest path in power/communication networks.
5. Simple dynamic programming problems.
6. Mini-project: Modeling an engineeing system as a graph problem.

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course

Essential Readings:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed., MIT Press, 2009.
2. J. Kleinberg and É. Tardos, *Algorithm Design*, 1st ed., Pearson Education, 2005.
3. E. Horowitz, S. Sahni, and S. Rajasekaran, *Fundamentals of Computer Algorithms*, 2nd ed., Universities Press, 2008.

Suggested Readings:

1. A. Levitin, *Introduction to the Design and Analysis of Algorithms*, 3rd ed., Pearson Education, 2012.
2. R. K. Shukla, *Analysis and Design of Algorithms: A Beginner's Approach*, Wiley India Pvt. Ltd., 2015.

REINFORCEMENT LEARNING (DSE-6 / GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Reinforcement Learning	4	3	0	1	NIL

Course Hours: L: 03, T: 00, P: 02

Course Objectives:

1. Introduce the mathematical and computational foundations of reinforcement learning for sequential decision-making problems.
2. Develop the ability to formulate computing tasks as MDPs/bandits and define reward signals, states, and actions precisely.
3. Enable students to implement and compare core RL algorithms for prediction and control in tabular settings.
4. Train students to build deep RL agents with function approximation, stable training pipelines, and reproducible experiments.
5. Develop competence in policy optimization and actor-critic methods for discrete and continuous control.
6. Explain how RL principles extend to preference-based learning and RLHF-style alignment used in instruction-following LLM pipelines (pretraining vs alignment fine-tuning).

Course Outcomes:

At the end of this course, students will be able to:

1. Explain RL formulations (MDP, return, policy, value functions) and derive Bellman equations for prediction and control.
2. Implement and evaluate tabular RL algorithms (DP, Monte Carlo, TD, SARSA, Q-learning) on standard toy environments.
3. Implement deep value-based RL with function approximation and evaluate stability using appropriate metrics and seeds.
4. Design and implement policy-gradient and actor-critic methods (including PPO-style objectives) for benchmark control tasks.
5. Apply preference-based learning concepts to build a simple reward model and use it to optimize a policy (RLHF-style loop at small scale).
6. Evaluate RL systems for robustness and risk (reward design, safety constraints, dataset shift in offline settings) and propose mitigations aligned with AI risk frameworks.

UNIT-I ***(11 hours)***

Foundations of Reinforcement Learning and MDP Modeling: agent–environment interaction and reward hypothesis, episodic vs continuing tasks and returns, Markov property and Markov decision processes (MDPs), policies and policy improvement idea, state-value and action-value functions, Bellman expectation and optimality equations, dynamic programming methods (policy evaluation, policy iteration, value iteration), Monte Carlo prediction and control, temporal-difference learning (TD(0), n-step intuition), exploration vs exploitation and ε-greedy strategies, on-policy vs off-policy learning, SARSA and Q-learning, contextual bandits as simplified RL.

UNIT-II ***(12 hours)***

Function Approximation and Deep Value-Based Reinforcement Learning: motivation for generalization in large state spaces, feature representations and linear function approximation, gradient-descent view of prediction, instability sources with bootstrapping and off-policy learning, experience replay and target networks, deep Q-networks (DQN) and loss design, overestimation bias and mitigation (Double Q intuition), architectural enhancements (dueling/value-advantage decomposition), exploration in deep RL (noise and schedules), practical evaluation methodology (learning curves, variance across random seeds, sample efficiency vs wall-clock efficiency), debugging signals and failure patterns (divergence, reward scaling, value blow-up).

UNIT-III ***(11 hours)***

Policy Optimization and Actor-Critic Methods: motivation for direct policy optimization, stochastic policies and log-likelihood gradients, REINFORCE with baselines and variance reduction, advantage functions and actor-critic architectures, entropy regularization for exploration, trust-region motivation and clipped surrogate objectives (PPO-style), continuous-action control and reparameterization intuition, off-policy actor–critic overview with replay buffers, practical training issues (normalization, stability, sensitivity to hyperparameters, monitoring and early stopping).

UNIT-IV ***(11 hours)***

RL Systems in Practice, Offline/Model-Based RL, and RLHF-Style Preference Learning: model-based RL overview (learning dynamics models, planning vs learning trade-offs), offline RL fundamentals, multi-agent RL basics (self-play, cooperation vs competition, non-stationarity), reward design and reward hacking, safe RL concepts, preference-based RL (pairwise comparisons, learning a reward model from preferences), RLHF-style alignment loop for instruction-following LLMs as an application of policy optimization and reward modeling (pretraining by next-token prediction followed by supervised instruction tuning and RLHF-style fine-tuning), KL-regularized optimization intuition for keeping a policy close to a reference model, reproducibility practices for RL systems (seeding, versioning, reporting).

Practical Component: ***(30 hours)***

1. Environment and tooling setup: Python RL stack, Gymnasium-style environment interaction, seeding for reproducibility, logging and experiment folder structure.
2. Implement multi-armed bandits (ε-greedy, UCB-style, Thompson-style) and compare regret on multiple reward distributions.
3. Implement a Gridworld MDP and solve it via policy evaluation + value iteration; visualize value functions and greedy policies.
4. Implement Monte Carlo prediction/control on an episodic environment; analyze variance and sensitivity to episode count.
5. Implement TD prediction (TD(0), n-step extension) and compare learning curves against Monte Carlo under identical seeds.
6. Implement SARSA (on-policy) and Q-learning (off-policy) for control; study the impact of exploration schedules and step sizes.
7. Build an evaluation harness: multiple seeds, confidence intervals/summary statistics, checkpointing, and standardized reporting plots.
8. Implement value-function approximation (linear or small neural network) and demonstrate at least one instability mode and a mitigation.

9. Implement DQN with replay buffer + target network on CartPole (or equivalent); add one improvement (e.g., Double DQN or dueling head).
10. Implement an actor–critic baseline (A2C-style) and then a PPO-style clipped objective; compare stability and sample efficiency.
11. Preference learning lab (RLHF-style at small scale): collect pairwise preferences over short trajectories or text outputs, train a reward model, then optimize a policy against the learned reward with a constraint to stay close to a reference policy (e.g., KL-penalty), and evaluate reward hacking risks.
12. Mini-project: choose a computing/system problem (game, scheduling, resource allocation, simulated control, or small text-policy task), define state/action/reward clearly, implement two algorithm families (value-based vs policy/actor–critic or RLHF-style preference loop), run ablations, and submit a reproducible report with code and logs.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.
2. M. Lapan, *Deep Reinforcement Learning Hands-On*, 2nd ed., Packt Publishing, 2020.
3. A. Zai and B. Brown, *Deep Reinforcement Learning in Action*, Manning Publications, 2020.
4. N. Lambert, *Reinforcement Learning from Human Feedback*, online book / PDF, 2026.

Suggested Readings:

1. Canonical papers / surveys
2. V. Mnih et al., “Human-Level Control through Deep Reinforcement Learning,” *Nature*, 2015.
3. J. Schulman et al., “Proximal Policy Optimization Algorithms,” arXiv:1707.06347, 2017.
4. P. Christiano et al., “Deep Reinforcement Learning from Human Preferences,” *Advances in Neural Information Processing Systems (NIPS)*, 2017.
5. L. Ouyang et al., “Training Language Models to Follow Instructions with Human Feedback,” 2022.
6. M. Alshiekh et al., “Safe Reinforcement Learning via Shielding,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
7. S. Gu et al., “A Review of Safe Reinforcement Learning: Methods, Theories, and Applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
8. S. Levine et al., “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems,” 2020.

BIG DATA ANALYTICS (DSE-7 / GE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Big Data Analytics	4	3	0	1	NIL

Course Objectives:

1. Understand fundamentals and ecosystem of Big Data.
2. Implement distributed storage and processing using Hadoop and MapReduce.
3. Explore NoSQL and data warehousing tools.
4. Apply analytics frameworks to extract insights from big data.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain Big Data concepts and tools.
2. Develop programs using Hadoop and MapReduce.
3. Use MongoDB, Hive, and Pig for big data processing.
4. Apply Spark for large-scale data analytics.

UNIT-I ***(12 Hours)***

Introduction to Big Data and Analytics: Classification of Data, Characteristics of Big Data, Evolution of Big Data, Traditional BI vs Big Data, Hadoop Ecosystem, Types of Analytics, NoSQL overview.

UNIT-II ***(12 Hours)***

Hadoop and MapReduce Framework: Hadoop architecture, HDFS, YARN, MapReduce programming model, Mapper, Reducer, Combiner, Partitioner, Searching and Sorting, Compression.

UNIT-III ***(12 Hours)***

NoSQL and Data Processing Tools: NoSQL databases, MongoDB architecture and query language, Hive architecture, HQL, File formats, User Defined Functions.

UNIT-IV ***(9 Hours)***

Big Data Analytics Frameworks and Applications: Pig architecture and Pig Latin, Pig vs Hive, Apache Spark architecture, Data analysis with Spark, Text and Web Analytics, PageRank, Case studies.

Practical Component: **(30 Hours)**

1. Installation and configuration of Hadoop ecosystem components and execution of HDFS file operations.
2. Development of MapReduce programs such as Word Count and Matrix Multiplication.

3. Implementation of CRUD (Create, Read, Update, Delete) and aggregation operations using MongoDB.
4. Writing and execution of data processing scripts using Hive and Pig.
5. Implementation of data analytics tasks using Apache Spark.
6. Mini-project involving the design and development of a Big Data Analytics application.

List of Experiments

Note: The course instructor will design experiments and mini-projects to complete the practical component of the course

Essential Readings:

1. S. Acharya and S. Chellappan, *Big Data and Analytics*, 1st ed., Wiley India, 2015.
2. R. Kamal and P. Saxena, *Big Data Analytics*, 1st ed., McGraw-Hill Education, 2017.
3. T. White, *Hadoop: The Definitive Guide*, 4th ed., O'Reilly Media, 2015.

Suggested Readings:

1. T. Erl, W. Khattak, and P. Buhler, *Big Data Fundamentals: Concepts, Drivers & Techniques*, 1st ed., Pearson Education, 2016.
2. A. Shook and D. Miner, *MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems*, 1st ed., O'Reilly Media, 2015.

SOFTWARE MAINTENANCE AND EVOLUTION (DSE-7/GE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Software Maintenance and Evolution	4	3	1	0	Software Engineering, Object Oriented Software Engineering

Course Objectives:

1. To introduce the concepts and importance of software maintenance and evolution.
2. To understand different types of software maintenance and their impact on cost and quality.
3. To study software evolution laws, models, and processes.
4. To analyze program comprehension, reengineering, and refactoring techniques.
5. To expose students to modern challenges in maintaining large and long-lived software systems.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain the role of software maintenance in the software life cycle.
2. Classify and analyze different types of software maintenance activities.
3. Apply software evolution models and laws to real software systems.
4. Analyze legacy systems and apply reengineering and refactoring techniques.
5. Evaluate maintenance strategies using quantitative and qualitative measures.

UNIT-I ***(09 hours)***

Introduction to Software Maintenance: Software maintenance: definition and need; maintenance as a phase of software life cycle; types of maintenance: corrective, adaptive, perfective, and preventive; cost of maintenance; maintenance process models; maintenance planning and management; maintenance standards and documentation.

UNIT-II ***(12 hours)***

Program Comprehension and Maintenance Techniques: Program understanding and reverse engineering; legacy systems and maintenance challenges; code analysis techniques; documentation recovery; impact analysis; change management; configuration management in maintenance; tools and environments for software maintenance.

UNIT-III ***(11 hours)***

Software Evolution: Software evolution concepts; Lehman's laws of software evolution; evolution processes and models; software aging and decay; evolution of requirements and architecture; managing continuous evolution; metrics for software evolution and maintainability.

UNIT-IV ***(13 hours)***

Reengineering, Refactoring, and Modern Maintenance Issues: Software reengineering: objectives and process; reverse engineering and forward engineering; code restructuring and refactoring techniques;

design recovery; migration and modernization of legacy systems; maintenance of object-oriented systems; maintenance challenges in agile, DevOps, and open-source environments; case studies.

Tutorial Component: ***(15 hours)***

1. Identification and classification of maintenance activities for a given software system.
2. Case study on cost and effort distribution in software maintenance.
3. Program comprehension exercise on a medium-sized codebase.
4. Impact analysis for proposed changes in an existing system.
5. Study and discussion of Lehman's laws using real software examples.
6. Analysis of software evolution metrics and maintainability indices.
7. Refactoring exercise: identifying code smells and proposing improvements.
8. Legacy system reengineering case study.
9. Critical review of maintenance challenges in agile or DevOps-based projects.

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Essential Readings:

1. T. M. Pigoski, *Practical Software Maintenance: Best Practices for Managing Your Software Investment*, IEEE Computer Society Press, 2001.
2. M. M. Lehman and J. F. Ramil, *Software Evolution*, Springer, 2006.
3. M. Fowler, *Refactoring: Improving the Design of Existing Code*, 2nd ed., Addison-Wesley, 2018.

Suggested Readings:

1. R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill Education, 2019.
2. Y. Singh and K. K. Aggarwal, *Software Engineering*, 2nd ed., New Age International Publishers, 2018.

DIGITAL CURRENCIES AND BLOCKCHAIN (DSE-7/GE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Currencies and Blockchain	4	3	0	1	Blockchain Essentials, Cryptography Essentials

Course Objectives:

1. To introduce concepts and the evolution of cryptocurrencies and blockchain systems.
2. To provide in-depth knowledge of blockchain architecture and cryptographic primitives.
3. To develop skills in cryptocurrency design, scripting, and transaction validation.
4. To understand consensus mechanisms and decentralized trust models.
5. To explore real-world applications, challenges, and future directions of blockchain technology.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand the evolution, principles, and significance of digital currencies and blockchain technologies.
2. Analyze blockchain architectures, consensus mechanisms, and cryptographic foundations.
3. Apply cryptographic and scripting techniques to design cryptocurrency systems.
4. Evaluate security, privacy, and trust issues in blockchain-based platforms.
5. Design and assess blockchain-based solutions for real-world applications.

UNIT-I ***(11 hours)***

Fundamentals of Digital Currency and Blockchain*:* Cryptocurrency - origin and evolution, importance and use cases, legal and regulatory aspects, blockchain structure and components, interaction between blockchain and cryptocurrencies, public, private, and consortium blockchains, and hardware and software requirements for blockchain systems.

UNIT-II ***(11 hours)***

Functional Aspects of Cryptocurrency: Bitcoin architecture and transaction lifecycle; distributed consensus and atomic broadcast; alternatives to Bitcoin consensus; Byzantine fault-tolerant mechanisms; alternative cryptocurrencies; blockchain-based cryptocurrency applications; technologies adopted in blockchain platforms.

UNIT-III ***(11 hours)***

Cryptographic Foundations and Bitcoin Scripting: Cryptographic hash functions; puzzle-friendly and collision-resistant hashes; digital signatures and public key cryptography; Bitcoin scripting language; transaction scripts; pay-to-public-key-hash (P2PKH); pay-to-script-hash (P2SH); Segregated Witness; hash time-locked contracts; atomic swaps.

UNIT-IV ***(12 hours)***

Security, Privacy, and Advanced Blockchain Applications: Security threats and vulnerabilities in blockchain systems; building secure cryptocurrency platforms; privacy-preserving mechanisms; zero-knowledge proofs; blockchain-based payment systems; smart contracts and decentralized applications; e-governance and enterprise blockchain solutions; emerging trends including DeFi, NFTs, and CBDCs.

Practical Component: ***(30 hours)***

1. Installation and configuration of the blockchain development environment.
2. Creation and management of cryptocurrency wallets.
3. Writing and deploying smart contracts.
4. Implementing token standards and transactions.
5. Case studies on blockchain platforms and real-world deployments.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. N. Daskalakis and P. Georgitseas, *An Introduction to Cryptocurrencies: The Crypto Market Ecosystem*, 1st ed., Routledge & CRC Press, 2020.
2. A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction,* Princeton University Press, 2016.
3. A. M. Antonopoulos and D. A. Harding, *Mastering Bitcoin: Programming the Open Blockchain*, 3rd ed., O'Reilly Media, 2023.

Suggested Readings:

1. M. Grabowski, *Cryptocurrencies: A Primer on Digital Money*, 1st ed., Routledge, 2019.
2. D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World,* Penguin, 2016.

MULTIUSER AND SOCIAL AR/VR (DSE-7/ GE-8)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Multiuser and Social AR/VR	4	3	0	1	Foundations of Augmented and Virtual Reality, Computer Networks

Course Objectives:

1. To understand the architecture required to host multiple users in a shared virtual environment.
2. To implement real-time synchronization of avatars, spatial audio, and interactive objects.
3. To explore the psychological aspects of social presence and "embodiment" in XR.
4. To address the unique security and moderation challenges in social AR/VR spaces.

Course Outcomes:

1. Develop networked XR applications using frameworks like Photon (PUN/Fusion), Mirror, or Normcore.
2. Optimize network traffic for smooth interactions in low-bandwidth scenarios.
3. Design expressive avatars with synchronized lip-sync and hand-tracking.
4. Evaluate user experience (UX) based on spatial social cues and telepresence metrics.

UNIT-I ***(15 hours)***

Foundations of Networked XR: Architecture for XR: Client-Server vs. Peer-to-Peer vs. Related Architectures; Introduction to "Authoritative Servers"; Synchronization Challenges: State synchronization, Remote Procedure Calls (RPCs), and Network Observables; Latency Mitigation: Dead Reckoning, Linear Interpolation (Lerp), and Client-Side Prediction; AR Cloud: Shared spatial anchors and persistent world-mapping for multi-user AR.

UNIT-II ***(12 hours)***

Social Presence and Avatar Systems: Representation: High-fidelity vs. Low-poly avatars; The "Uncanny Valley" in social VR.; Embodiment: IK (Inverse Kinematics) for body tracking; Synchronizing head and hand movements.; Social Cues: Eye tracking, facial expression mapping, and procedural lip-syncing; Spatial Audio: 3D sound attenuation, HRTF (Head-Related Transfer Functions), and voice-chat occlusion.

UNIT-III ***(08 hours)***

Shared Interaction and Physics: Object Ownership: Handling collisions and interactions when multiple users grab the same object.; Concurrency Control: Locking mechanisms for shared UI and 3D assets; Multi-user AR: Collaborative design, remote assistance (See-What-I-See), and shared holograms; Navigation: Synchronized teleportation vs. continuous movement; Preventing motion sickness in shared spaces.

UNIT-IV ***(10 hours)***

Ethics, Privacy, and Scalability: **Social Safety:** Personal space bubbles, hand-gesture filtering, and real-time moderation tools; Privacy: Data security in spatial computing; Risks of biometric tracking (eye/gaze data); Massive Multiuser XR: Techniques for sharding and interest management (NetCulling); Case Studies: Analyzing VRChat, Rec Room, Horizon Worlds, and Microsoft Mesh.

Practical Component: ***(30 hours)***

1. Basic Connection: Set up a "Network Manager" to allow two users to join a shared VR room as simple primitive shapes.
2. Avatar Sync: Implement a full-body avatar that synchronizes the movement of a VR user's head and hands to remote players.
3. Spatial Voice Chat: Integrate a Voice-over-IP (VoIP) system where volume decreases as users move away from each other in 3D space.
4. Shared Interaction: Create a "Collaborative Whiteboard" where multiple users can draw or move 3D objects simultaneously.
5. Persistent AR: Use Azure Spatial Anchors or Niantic Lightship to place a digital object in a physical room that remains visible to multiple mobile AR users.
6. Safety Zone Logic: Script a "Proximity Bubble" that makes other players invisible/transparent if they get too close to the user's avatar.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. B. Shneiderman, *Human-Centered AI*, Oxford University Press, 2022.
2. J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality,* Morgan & Claypool Publishers, 2015
3. T. Parisi, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*, O'Reilly Media, 2015.

Suggested Readings:

1. Unity Technologies, *Unity Multiplayer Networking Documentation (Netcode for GameObjects / Fusion)*, Unity Manual, latest edition.
2. M. Ball, *The Metaverse: And How It Will Revolutionize Everything,* Liveright Pub Corp, 2022

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of CSE in SEMESTER VIII

LOW LEVEL SOFTWARE SYSTEM (GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Low Level Software System	4	3	0	1	Data Structures, Fundamentals of Operating Systems

Course Objectives:

1. Introduce system-level thinking bridging hardware and software.
2. Provide foundational understanding of firmware, kernel, and user-space interaction.
3. Develop awareness of performance and memory behavior in modern processors.
4. Familiarize students with real-time and embedded system considerations.
5. Introduce basic low-level security mechanisms in modern systems.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain how software interacts with hardware from boot process to application execution.
2. Understand basic kernel responsibilities including interrupts, scheduling, and memory management.
3. Analyze performance implications of low-level programming decisions.
4. Apply fundamental concepts of synchronization, real-time constraints, and system security.
5. Interpret compilation, linking, and execution flow at a system level.

UNIT-I ***(10 hours)***

Bootstrapping and Control Transfer: System Boot Process: Firmware concepts (BIOS/UEFI, U-Boot overview), Secure Boot and basic trust concepts, Memory Layout Basics: Stack, Heap, Data, BSS, Introduction to MMU and virtual memory, Binary formats (ELF) and program loading, Introduction to HAL (Hardware Abstraction Layer),Calling conventions and stack fundamentals

UNIT-II ***(13 hours)***

Kernel Runtime Evolution and Application Integration: Kernel roles and execution context, Basics of interrupts and interrupt handling, Timers and system clock concepts, Introduction to device drivers and I/O subsystems, System calls and user–kernel boundary, Overview of process creation and program execution, The exec Pipeline, Embedded vs general-purpose OS (Linux, FreeRTOS overview).

UNIT-III *(12 hours)*

Microarchitectural Performance, Memory Models, and Determinism: Hardware Atomics, Compare-and-Swap (CAS) vs. Load-Link/Store-Conditional (LL/SC), Memory Consistency Models, Cache Coherency Interaction, The Pipeline-Software Interface, Instruction-Level Parallelism (ILP), Synchronization problems, Real-time systems: Hard vs Soft real-time, Scheduling basics: Rate Monotonic and EDF (conceptual overview), Introduction to Worst-Case Execution Time (WCET)

UNIT-IV ***(10 hours)***

Systems Toolchains, Optimization, and Low-Level Security: Compilation process, Static vs Dynamic linking (conceptual understanding), Compiler optimizations overview (-O levels), Cache-aware programming basics, Stack protection and memory protection concepts, Introduction to ASLR and DEP, Overview of Overview of speculative execution vulnerabilities (Spectre and Meltdown),Secure coding principles in system software.

Practical Component: ***(30 hours)***

1. Analyze memory layout using C programs and tools (objdump, nm).
2. Study stack frames and calling conventions via simple assembly inspection.
3. Writing and analyzing interrupt-driven programs (simulation or embedded board).
4. Basic driver-style interaction (GPIO or simulated I/O).
5. Measuring execution time and observing compiler optimization effects.
6. Implementing simple synchronization examples using threads.
7. Mini-project: System-level performance or synchronization case study.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. J. L. Hennessy, D. A. Patterson, and C. Kozyrakis, *Computer Architecture: A Quantitative Approach*, 7th ed., Morgan Kaufmann, 2022.
2. D. Lacamera, *Embedded Systems Architecture: Explore architectural concepts, pragmatic design patterns, and best practices to produce robust systems*, 2nd ed., Packt Publishing, 2021.
3. R. E. Bryant and D. R. O'Hallaron, *Computer Systems: A Programmer's Perspective*, 3rd ed., Pearson Education, 2016.
4. R. H. Arpaci-Dusseau and A. C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, Version 1.10, Arpaci-Dusseau Books, 2018.

Suggested Readings:

1. D. P. Bovet and M. Cesati, *Understanding the Linux Kernel*, 3rd ed., O'Reilly Media, 2005.
2. A. S. Tanenbaum and H. Bos, *Modern Operating Systems*, 5th ed., Pearson, 2022.
3. R. Jain, *The Art of Computer Systems Performance Analysis*, Wiley, 1991.
4. G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 4th ed., Springer, 2020.
5. Official white papers on Spectre and Meltdown vulnerabilities (Intel, ARM, and related security advisories).
6. Processor ISA manuals (e.g., Intel® 64 and IA-32 Architectures Software Developer's Manual, ARM Architecture Reference Manual) and ELF specification documents.

OOP CONCEPTS (GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
OOP Concepts	4	3	0	1	Fundamentals of Computer Programming

Course Objectives:

1. Introduce fundamental concepts of object-oriented programming and design.
2. Enable students to model real-world problems using classes and objects.
3. Provide hands-on experience with object-oriented languages and tools.
4. Promote good programming practices using modularity, reuse, and maintainability.
5. Familiarize students with exception handling, file I/O, and generic programming techniques

Course Outcomes:

At the end of this course, students will be able to:

1. Understand core object-oriented concepts such as abstraction, encapsulation, inheritance, and polymorphism.
2. Design and implement object-oriented solutions using classes, objects, and interfaces.
3. Apply principles of inheritance and polymorphism to develop reusable and extensible software.
4. Implement exception handling and file handling mechanisms for robust applications.
5. Use object-oriented design principles to model real-world systems.
6. Develop complete object-oriented applications using Java / C++.

UNIT-I ***(12 Hours)***

Introduction to Object Oriented Programming: Programming Paradigms; Procedural vs Object-Oriented Programming; Objects, Classes, Data Abstraction, Encapsulation; Benefits of OOP; Java/C++ Basics; Control Statements; I/O Operations.`

UNIT-II ***(10 Hours)***

Classes, Objects and Constructors: Class definition; Access Specifiers; Constructors and Destructor; Constructor Overloading; this pointer; Static Members; Memory Allocation.

UNIT-III ***(12 Hours)***

Inheritance and Polymorphism: Types of Inheritance; Method Overriding; Polymorphism; Function and Operator Overloading; Virtual Functions; Abstract Classes; Interfaces.

UNIT-IV ***(11 Hours)***

Advanced OOP Concepts and Applications: Exception Handling; Generic programming- Templates in C++.

Practical Component: *(30 Hours)*

1. Development of programs demonstrating classes and objects.
2. Implementation of constructors and destructors in object-oriented programs.
3. Programs illustrating inheritance and polymorphism concepts.
4. Implementation of exception handling mechanisms.
5. File handling programs for reading and writing data.
6. Mini project applying object-oriented design principles.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. E. Balagurusamy, *Object Oriented Programming with Java*, 6th ed., McGraw Hill Education, 2019.
2. H. Schildt, *Java: The Complete Reference*, 11th ed., McGraw Hill Education, 2018.
3. R. Lafore, *Object-Oriented Programming in C++*, 4th ed., Sams Publishing, 2001.

Suggested Readings:

1. B. McLaughlin, G. Pollice, and D. West, *Head First Object-Oriented Analysis and Design*, 1st ed., O'Reilly Media, 2006.
2. R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, 1st ed., Prentice Hall, 2008.
3. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed., Addison-Wesley, 1994.

GENERATIVE AI (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Generative AI	4	3	0	1	NIL

Course Objectives:

1. Introduce the mathematical and conceptual foundations of generative modeling.
2. Enable hands-on experience with modern generative architectures.
3. Provide practical exposure to training and fine-tuning large generative models.
4. Familiarize students with prompt engineering and alignment techniques.
5. Develop critical understanding of safety, bias, and responsible AI usage.
6. Prepare students for research and industry applications of Generative AI.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand the theoretical foundations of generative modeling and representation learning.
2. Implement and train generative models such as autoencoders, VAEs, GANs, and diffusion models.
3. Apply large language models (LLMs) and transformer architectures for text generation and understanding.
4. Develop multimodal generative systems for text, image, and audio data.
5. Evaluate generative models using qualitative and quantitative metrics.
6. Analyze ethical, legal, and societal implications of Generative AI.

UNIT-I ***(10 hours)***

Foundations of Generative Modeling: Discriminative vs Generative Models; Probability Distributions and Likelihood-Based Learning; Maximum Likelihood Estimation; Latent Variable Models; Representation Learning; Autoencoders: Architecture, Training, and Limitations; Introduction to Variational Inference.

UNIT-II ***(13 hours)***

Variational Autoencoders and GAN: Variational Autoencoders (VAEs): Evidence Lower Bound (ELBO), Reparameterization Trick; Conditional VAEs; Generative Adversarial Networks (GANs): Minimax Objective, Training Dynamics, Mode Collapse; Variants: DCGAN, WGAN, CycleGAN; Evaluation Metrics: Inception Score, FID.

UNIT-III ***(12 hours)***

Diffusion Models and Transformers: Denoising Diffusion Probabilistic Models (DDPMs); Score-Based Generative Models; Sampling and Acceleration Techniques; Transformer Architecture: Self-Attention, Positional Encoding; Large Language Models (LLMs): Pretraining, Fine-Tuning, Instruction Tuning; Prompt Engineering and In-Context Learning.

UNIT-IV ***(10 hours)***

Multimodal Generative AI and Applications: Text-to-Image Models (e.g., DALL·E-style architectures); Image-to-Image Translation; Speech and Audio Generation; Multimodal Transformers; Retrieval-Augmented Generation (RAG); Applications in Content Creation, Healthcare, Education, and Scientific Research; Ethical, Legal, and Social Issues: Bias, Hallucination, Copyright, and AI Safety.

Practical Component: ***(30 hours)***

1. Implementing autoencoders and VAEs for image generation.
2. Training GANs for image synthesis tasks.
3. Implementing diffusion models for denoising and generation.
4. Fine-tuning transformer-based language models for text generation tasks.
5. Designing multimodal generative pipelines (text-to-image or image captioning).
6. Mini-project on a real-world Generative AI application.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
2. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. (draft), Pearson, 2023.
3. I. Goodfellow, “NIPS 2016 Tutorial: Generative Adversarial Networks,” arXiv:1701.00160, 2016.

Suggested Readings:

1. D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” *International Conference on Learning Representations (ICLR)*, 2014.
2. A. Vaswani et al., “Attention Is All You Need,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
3. J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

SCIENTIFIC MACHINE LEARNING (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Scientific Machine Learning	4	3	0	1	NIL

Course Objectives:

1. Introduce students to the principles of Scientific Machine Learning, combining mechanistic models with machine learning.
2. Enable students to solve ODEs and PDEs using neural networks without traditional discretization methods.
3. Familiarize students with Physics-Informed Neural Networks (PINNs) and constrained learning frameworks.
4. Provide hands-on experience with operator learning and surrogate modeling.
5. Train students to handle uncertainty, interpretability, and robustness in scientific data-driven models.
6. Encourage interdisciplinary thinking through case studies and mini-projects.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand the foundations of Scientific Machine Learning and its role in integrating physical laws with data-driven models.
2. Apply machine learning techniques to scientific and engineering problems involving noisy, sparse, and high-dimensional data.
3. Develop Physics-Informed Neural Networks (PINNs) for solving ordinary and partial differential equations.
4. Implement operator learning approaches such as DeepONets and Fourier Neural Operators for learning solution mappings.
5. Use SciML techniques for real-world applications in physics, healthcare, climate science, and engineering systems.
6. Critically evaluate model interpretability, generalization, and uncertainty in scientific ML applications.

UNIT-I ***(10 hours)***

Introduction to Scientific Machine Learning: Scientific Computing vs Data-Driven Modeling; Motivation for Scientific Machine Learning; Limitations of Purely Data-Driven ML; Role of Physical Laws, Constraints, and Conservation Principles; Overview of Differential Equations in Science and Engineering; Introduction to Hybrid Modeling; Examples from Fluid Dynamics, Biomedical Engineering, Climate Modeling, and Materials Science.

UNIT-II ***(13 hours)***

Machine Learning for Scientific Data: Scientific Data Characteristics: Noise, Sparsity, and Multiscale Nature; Regression and Classification for Scientific Applications; Gaussian Processes for Function Approximation; Kernel Methods; Dimensionality Reduction Techniques: PCA, Autoencoders; Model Generalization and Extrapolation in Scientific Domains; Bias–Variance Tradeoff in Physics-Based Systems.

UNIT-III ***(10 hours)***

Physics-Informed Neural Networks (PINNs): Neural Networks as Function Approximators; Automatic Differentiation; Formulation of Physics-Informed Loss Functions; Solving Ordinary Differential Equations (ODEs) using PINNs; Solving Partial Differential Equations (PDEs); Boundary and Initial Conditions Handling; Training Challenges and Optimization Strategies; Case Studies: Burgers' Equation, Heat Equation, and Navier–Stokes Equations.

UNIT-IV ***(12 hours)***

Advanced SciML Topics and Applications: Operator Learning: Deep Operator Networks (DeepONets), Fourier Neural Operators (FNOs); Reduced-Order Modeling and Surrogate Models; Uncertainty Quantification and Bayesian PINNs; Scientific Discovery using ML (Symbolic Regression and Equation Discovery); Ethical and Reliability Considerations in Scientific AI; Case Studies: Digital Twins, Medical Imaging, Climate Prediction, and Blood Flow Modeling.

Practical Component: ***(30 hours)***

1. Implementing regression and Gaussian Process models for scientific datasets.
2. Solving ODEs using neural networks and automatic differentiation.
3. Implementing Physics-Informed Neural Networks (PINNs) for benchmark PDEs.
4. Training PINNs for boundary value and initial value problems.
5. Implementing operator learning models such as DeepONets or Fourier Neural Operators.
6. Developing a mini-project applying SciML to a real-world scientific problem.
7. Developing a natural language processing pipeline for text classification, sentiment analysis, or spam detection.
8. Building and testing generative models (e.g., GANs) for image generation tasks.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-Informed Machine Learning," *Nature Reviews Physics*, vol. 1, pp. 1–19, 2021.
2. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

Suggested Readings:

1. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
2. R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018

VISUAL COMPUTING (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Visual Computing	4	3	0	1	Probability and Statistics for Computer Science

Course Objectives:
1. Introduce fundamental concepts of visual computing and image formation.
2. Understand computer vision and graphics techniques for visual data processing.
3. Apply algorithms for image analysis, feature extraction, and visualization.
4. Explore real-world applications of visual computing systems.

Course Outcomes:
1. Explain core principles of visual computing and image representation.
2. Implement basic image processing and computer vision algorithms.
3. Analyze visual data for recognition and interpretation tasks.
4. Apply visual computing techniques to real-world applications.

UNIT-I ***(12 hours)***
Introduction to Visual Computing: Overview of Visual Computing, Image Formation, Image Representation, Color Models, Human Visual System, Digital Image Fundamentals.

UNIT-II ***(12 hours)***
Image Processing Techniques: Image enhancement, filtering, edge detection, segmentation, morphological operations, frequency domain analysis.

UNIT-III ***(12 hours)***
Computer Vision and Feature Analysis: Feature detection and description, object detection, motion analysis, camera models, stereo vision, Overview of 3D reconstruction

UNIT-IV ***(9 hours)***
Advanced Topics and Applications: Visual learning and deep vision (overview), graphics pipelines, visualization techniques, applications in AR/VR, medical imaging, autonomous systems and smart surveillance.

Practical Component: ***(30 hours)***
1. Implementation of programs for image representation and geometric transformations such as scaling, rotation, and translation.
2. Development of image filtering and enhancement techniques using spatial and frequency domain methods.

3. Implementation of feature extraction and feature matching algorithms.
4. Development of basic object detection and tracking applications.
5. Mini-project involving the design and implementation of a visual computing application.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022.
2. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018.
3. D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 2nd ed., Pearson, 2011.

Suggested Readings:

1. A. Watt, *3D Computer Graphics*, 3rd ed., Addison-Wesley, 2000.
2. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016

‘

APPLIED AI (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Applied AI	4	3	0	1	Fundamentals of Data Analytics

Course Objectives:

1. Provide a structured understanding of AI techniques tailored to specific data types: numeric, text, image, audio, video, and time-series.
2. Develop skills in preprocessing, feature engineering, and modeling for different data modalities.
3. Enable students to select, design, and implement appropriate AI/ML pipelines based on data characteristics.
4. Foster the ability to build end-to-end data analytics solutions using real-world datasets.
5. Prepare students for industry roles in AI engineering, data science, and analytics.

Course Outcomes:

On completion of this course, students will be able to:

1. To understand how AI techniques are applied to different types of data in real-world analytics.
2. To build competency in selecting appropriate AI models based on data characteristics.
3. To apply AI techniques for numeric, text, multimedia, and time-series data.
4. To develop analytical, modeling, and decision-making skills using AI-driven approaches.

UNIT-I ***(12 hours)***

AI for Numeric and Structured Data: Introduction to Data Types in AI (Numeric, Text, Audio/Video, Images, Time series), Characteristics of Numeric and Structured Data, Data Collection and Quality issues, Data Preprocessing, Handling missing values, Data Normalization and Standardization, Outlier Detection and Treatment, Exploratory Data Analysis (EDA), Data Visualization Techniques, ML for Numeric Data: Linear and Multiple Regression, Logistic Regression, k-Nearest Neighbour, Decision Trees and Ensemble Methods (Random Forest), Evaluation Metrics (Accuracy, Precision, Recall, RMSE).

UNIT-II ***(10 hours)***

AI for Text and Natural Language Data: Introduction to Unstructured and Textual data: Challenges and Opportunities, Text Preprocessing Techniques (Tokenization, Stop-word removal, Stemming and Lemmatization), Text Representation Methods: Bag-of-Words, TF-IDF, Word Embeddings (Word2Vec, GloVe), Introduction to Natural Language Processing: Tasks and Applications, Sentiment Analysis, Topic Modeling, Text Classification, Named Entity Recognition, Transformers and BERT Architecture, Fine-tuning Pre-Trained Language Models for Downstream Tasks.

UNIT-III ***(11 hours)***

AI for Image, Audio, and Video Analytics: Introduction to Multimedia Data, Digital Image Representation, Image Processing and Enhancement, Feature Extraction, Fundamentals of Computer Vision, Convolutional Neural Networks (CNN) Architecture, CNN Applications in Image Analytics, Object Detection and Image Classification, Audio Data Analytics, Audio Signal Representation, Feature Extraction from Audio Data, Audio Classification and Speech Recognition Basics, Introduction to Video Analytics (Frame Extraction, Motion Analysis, Video Classification).

UNIT-IV ***(10 hours)***

AI for Time-series and Sequential Data Analytics: Introduction to Time-Series and Sequential Data, Characteristics and Challenges of Temporal Data, Time-series Data Processing and Visualization, Trend, Seasonality, and Noise Analysis, Feature Engineering for Time-dependent Data, Traditional Time-series Models, Moving Average Models, Exponential Smoothing, ARIMA, ML techniques for Time-series Forecasting (Feature-based Regression, XGBoost), Deep Learning Models for Sequential Data, RNNs, LSTM Networks, Anomaly Detection in Time-series Data.

Practical Component: ***(30 hours)***

1. Data preprocessing and EDA on structured datasets, including handling missing values, normalization, and visualization.
2. Implementation of machine learning models for numeric data (Linear Regression, Logistic Regression, k-NN, Decision Trees, Random Forest) with performance evaluation.
3. Text preprocessing and representation using Bag-of-Words, TF-IDF, and word embeddings; implementation of text classification or sentiment analysis.
4. Fine-tuning of a pre-trained Transformer-based model (e.g., BERT) for a downstream NLP task.
5. Image classification using CNNs and basic object detection using deep learning frameworks.
6. Audio data preprocessing and feature extraction (e.g., MFCCs) with implementation of a simple audio classification model.
7. Time-series visualization and forecasting using ARIMA and machine learning models (e.g., regression/XGBoost).
8. Implementation of LSTM-based sequence models for time-series prediction or anomaly detection.
9. Mini-project involving an end-to-end AI pipeline on a real-world dataset (numeric, text, multimedia, or time-series), including preprocessing, modeling, evaluation, and reporting.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O'Reilly Media, 2022.
2. J. D. Kelleher, *Deep Learning*, MIT Press, 2019.
3. S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed., Packt Publishing, 2019.
4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

Suggested Readings:

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
2. D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed., Pearson, 2014.
3. A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*, O'Reilly Media, 2016.

PROGRAM ANALYSIS AND VERIFICATION (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Program Analysis and Verification	4	3	0	1	Data Structures

Course Objectives:

1. To introduce the mathematical and logical foundations required for program analysis and verification.
2. To study classical dataflow analysis techniques and pointer analysis for imperative programs.
3. To understand abstract interpretation as a principled framework for sound program analysis.
4. To introduce formal reasoning, automated verification techniques, and systematic program testing methods.
5. To develop the ability to reason about program correctness, safety, and behavior using formal and semi-formal methods.

Course Outcomes:

After successful completion of the course, the student will be able to:

1. Apply lattice theory and logical formalisms to reason about program properties.
2. Analyze programs using dataflow analysis techniques such as constant propagation, reaching definitions
3. Explain and apply abstract interpretation concepts, including abstract domains, Galois connections, and widening/narrowing, to approximate program semantics.
4. Reason about program correctness using Hoare logic, weakest preconditions, and model checking techniques.
5. Use automated verification approaches, including SMT/SAT solving.

UNIT-I ***(8 hours)***

Foundational Topics: Partial orders and lattices: Reflexivity, antisymmetry, transitivity, Least upper bound (join), greatest lower bound (meet). Propositional Logic.

UNIT-II ***(10 hours)***

Dataflow Analysis: Program Representation. Dataflow Analysis: Constant Propagation, Reaching Definitions, Available Expressions, Kildall's algorithm

UNIT-III ***(12 hours)***

Abstract Interpretation: Abstract domains and lattices, Abstract join-over-all-paths analysis, Galois connections and abstraction/concretization functions, Abstract interpretation as approximation of concrete semantics. Numerical abstract domains: intervals, affine equalities. Widening and narrowing.

UNIT-IV ***(15 hours)***

Program Correctness and Verification: Floyd-Hoare Logic, Weakest Precondition, Software Model Checking: symbolic execution, state-space reduction, state-less model checking, counter-example guided abstraction refinement (CEGAR), Introduction to SMT and Solvers: SAT, SMT, DPLL.

Essential Readings:

1. A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd ed., Pearson, 2006.
2. Jeannet, B. and A. Miné, Apron: (A) Library of Numerical Abstract Domains for Static Analysis. Lecture Notes in Computer Science (LNCS). Springer. 2009.
3. E. M. Clarke, O. Grumberg, and D. A. Peled. Model Checking, MIT Press, 1999.

Suggested Readings:

1. F. Nielson, H. R. Nielson, and C. Hankin, Principles of Program Analysis, Springer, 1999.
2. Cousot, R. and Cousot, P., Static Analysis Frameworks. Lecture Notes in Computer Science (LNCS), Springer, 1997.
3. D. Kroening and O. Strichman, *Decision Procedures:* An Algorithmic Point of View, 2nd ed., Springer, 2016.
4. B. C. Pierce et al., Software Foundations: Programming Language Foundations, Electronic textbook, 2018

Practical Component: ***(30 hours)***

1. Program representation and control-flow analysis exercises using static analysis tools (e.g., construction of Control Flow Graphs and intermediate representations).
2. Implementation and experimentation with classical dataflow analyses such as Constant Propagation, Reaching Definitions, and Available Expressions using compiler or analysis frameworks.
3. Development and evaluation of abstract interpretation–based analyses using numerical abstract domains (e.g., interval analysis) with widening and narrowing strategies.
4. Formal specification and verification of simple programs using Floyd–Hoare logic and weakest precondition reasoning.
5. Program verification using model checking tools, including symbolic execution and counterexample analysis.
6. Hands-on exercises with SAT/SMT solvers for constraint solving and automated reasoning tasks.
7. Mini-project involving static analysis or formal verification of a small software system using appropriate tools and frameworks.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

EMPIRICAL SOFTWARE ENGINEERING (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Empirical Software Engineering	4	3	0	1	Software Engineering, Object Oriented Software Engineering

Course Objectives:

1. To understand fundamental concepts and importance of empirical software engineering in practice.
2. To study software metrics, basic experimental design, and data collection techniques
3. To analyze software repositories and datasets for deriving useful insights.
4. To apply basic statistical and machine learning techniques for software data analysis.
5. To develop skills for conducting structured literature reviews and presenting empirical findings.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain basic concepts, processes, and ethical aspects of empirical software engineering.
2. Design simple empirical studies including formulation of research questions and data collection.
3. Analyze software metrics and repository data using appropriate tools and techniques.
4. Apply statistical and basic machine learning methods for data analysis and interpretation.
5. Prepare structured reports based on empirical studies and literature reviews.

UNIT-I ***(11 hours)***

Foundations of Empirical Software Engineering & Systematic Reviews: Introduction to empirical software engineering: concepts, motivation, and types of empirical studies; Empirical research process, study design, and , ethical considerations; Systematic Literature Reviews (SLR): planning, conducting, data extraction, synthesis, and reporting.

UNIT-II ***(11 hours)***

Software Metrics and Experimental Design*:*Fundamentals of software measurement and metrics (size, quality, object-oriented, and evolution metrics); Validation and industrial relevance of software metrics; Experimental design in software engineering: research questions, variables, hypothesis formulation, data collection, and selection of analysis techniques.

UNIT-III ***(10 hours)***

Mining Software Repositories and Data Sources*:* Introduction to mining software repositories (MSR); Version control systems, bug tracking systems, and configuration management data; Data extraction, static code analysis, and historical software analysis; Use of open research datasets for empirical studies.

UNIT-IV ***(13 hours)***

Data Analysis, Model Development, and Research Reporting: Statistical analysis of software engineering data and hypothesis testing; Model development using statistical and machine learning techniques; Model evaluation, cross-validation, and performance metrics; Text mining applications in software engineering; Threats to validity, countermeasures, ethical reporting, and research documentation.

Practical Component: ***(30 hours)***

1. Systematic literature review exercises including research question formulation and structured reporting
2. Software metrics computation and validation using statistical analysis techniques.
3. Mining and analysis of data from software repositories such as Git and bug tracking systems
4. Statistical hypothesis testing and empirical data analysis using appropriate tools.
5. Development and evaluation of fault prediction models using statistical and machine learning techniques.
6. Model comparison and interpretation using performance evaluation metrics and cross-validation.
7. Text mining and classification of defect reports using feature extraction and ML methods.
8. Mini-project involving end-to-end empirical study including data collection, analysis, modeling, validation, and research reporting.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. R. Malhotra, *Empirical Research in Software Engineering: Concepts, Analysis, and Applications*, CRC Press / Taylor & Francis, 2016.
2. C. Wohlin et al., *Experimentation in Software Engineering*, 2nd ed., Springer, 2012.
3. T. Menzies, L. Williams, and T. Zimmermann (eds.), *Perspectives on Data Science for Software Engineering*, Morgan Kaufmann, 2016.

Suggested Readings:

1. F. Shull, J. Singer, and D. I. K. Sjøberg (eds.), *Guide to Advanced Empirical Software Engineering*, Springer, 2008.
2. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed., Morgan Kaufmann, 2016.

CYBER LAW AND DIGITAL FORENSICS (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cyber Law and Digital Forensics	4	3	0	1	Cryptography Essentials

Course Objectives:

1. To understand security issues in the network and application layers.
2. To learn concepts related to cybercrime, cyber law, and digital investigations.
3. To acquire knowledge of forensic tools and techniques for evidence analysis.
4. To develop skills for the investigation and reporting of cybercrime incidents.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand various cybercrimes and cyber laws.
2. Analyze digital evidence and perform forensic investigations.
3. Apply forensic tools and techniques to identify and trace cyber attackers.
4. Interpret digital forensic findings for legal and investigative purposes

UNIT-I ***(11 hours)***

Introduction to Cyber Law and Cyber Crimes*:* Internet and cyberspace; cybercrime and its classification; hacking, cracking, viruses, malware, and cyber-attacks; pornography and intellectual property crimes; legal framework of cyber laws; Information Technology Act; social engineering; mail bombing; bug exploits; ethical and legal issues in cyber investigations.

UNIT-II ***(11 hours)***

Forensic Science and Investigation Process: Principles of digital forensics; scientific approach to forensic investigation; identification and classification of digital evidence; evidence acquisition and preservation; chain of custody; evidence handling and storage; forensic investigation methodology; documentation and reporting.

UNIT-III ***(11 hours)***

Digital and Network Forensics: Hardware forensics; file systems and hidden data; anti-forensics techniques; network forensics; analysis of network traffic; virtual systems and virtualization forensics; mobile device forensics; watermarking techniques; secure computation protocols

UNIT-IV ***(12 hours)***

Application Forensics and Cybercrime Investigation: Application forensics; email forensics; social media and cloud forensics; investigation of cybercrimes; forensic tools and techniques; cybercrime case studies; preparation of forensic reports; presentation of digital evidence in legal proceedings.

Practical Component: ***(30 hours)***

1. Identification and analysis of cybercrime incidents using real-world case studies.
2. Acquisition and preservation of digital evidence following standard forensic procedures.
3. File system analysis, including recovery of deleted files and examination of metadata.
4. Disk imaging and verification using forensic tools.
5. Analysis of system logs, registry entries, and user activity artifacts.
6. Network traffic capture and analysis for identifying malicious activity.
7. Email, web, and social media forensic investigation.
8. Basic malware and suspicious file analysis in a controlled environment.
9. Preparation of forensic investigation reports with proper documentation and legal compliance.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. B. Nelson, A. Phillips, and C. Steuart, *Guide to Computer Forensics and Investigations*, 6th ed., Cengage Learning, 2020.
2. J. R. Vacca, *Computer Forensics: Computer Crime Scene Investigation*, 2nd ed., Charles River Media, 2005.
3. N. Jain and D. R. Kalbande, *Digital Forensic: The Fascinating World of Digital Evidence*, 1st ed., Wiley, 2016.

Suggested Readings:

1. C. Easttom, *Computer Security Fundamentals*, 4th ed., Pearson, 2019.
2. M. Britz, *Computer Forensics and Cyber Crime : An Introduction*, 4th ed., Pearson, 2020.

CRYPTANALYSIS & SECURITY MODELS (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cryptanalysis & Security Models	4	3	0	1	Cryptography Essentials

Course Outcomes:
At the end of this course, students will be able to:

1. Understand classical and modern cryptanalysis techniques and their impact on cryptographic algorithms.
2. Analyze security models and adversarial assumptions.
3. Evaluate cryptographic protocol security using formal models.
4. Apply cryptanalysis methods to assess system security.
5. Identify security vulnerabilities and propose appropriate countermeasures.

Course Objectives:

1. Provide a thorough understanding of cryptanalysis techniques and security breaches.
2. Examine formal models of security and security proofs for cryptographic protocols.
3. Study security adversary models, attack strategies, and countermeasures.
4. Introduce security model frameworks used to analyze modern protocols.

UNIT-I ***(11 hours)***
Introduction to Cryptography & Cryptanalysis: Security threats, attacks, and models in network security; Security attacks, services and mechanisms; OSI security architecture and security policies; Cryptography basics; Classical encryption techniques - substitution and transposition ciphers; Introduction to cryptanalysis - ciphertext-only, known-plaintext, chosen-plaintext, chosen-ciphertext attacks; Goals of cryptanalysis.

UNIT-II ***(11 hours)***
Design and Cryptanalysis of Symmetric Ciphers: Block cipher design principles; Differential cryptanalysis; Linear cryptanalysis; Other modern attacks (integral, meet-in-the-middle); Modes of operation and their security implications; Security definitions - semantic security, IND-CPA, IND-CCA; Security of modern symmetric primitives and provable security concepts as applied to block ciphers and stream ciphers.

UNIT-III ***(11 hours)***
Public-Key Cryptanalysis & Security Models: Asymmetric cryptography overview: RSA, Diffie-Hellman; Adversarial models for public-key systems; Security definitions for public-key encryption schemes; Chosen-plaintext and chosen-ciphertext security models; Introduction to provable security and simulation-based security; Universal Composability (UC) security model basics.

UNIT-IV ***(12 hours)***

Security Protocols & Formal Security Analysis: Security protocol basics; Model of protocol execution; Authentication and key-exchange protocol analysis; Security proof techniques; Security protocols: SSL/TLS, IPSec, Kerberos, PGP; Security architectures and models; Countermeasures and modern cryptographic defenses such as elliptic curve cryptography.

Practical Component: ***(30 hours)***

1. Implementation of classical ciphers and basic cryptanalysis (e.g., substitution, transposition).
2. Differential and linear cryptanalysis simulations on test ciphers.
3. Evaluation of symmetric cipher security using IND-CPA and IND-CCA models.
4. Analysis of RSA and Diffie-Hellman under various attack models (chosen-plaintext / ciphertext).
5. Hands-on protocol security evaluation (SSL/TLS, IPSec).
6. Security tool experiments (e.g., hash collision tests, brute-force simulations).

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. W. Stallings, *Cryptography and Network Security: Principles and Practice*, 8th ed., Pearson, 2023.
2. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., CRC Press, 2020.
3. D. R. Stinson and M. Paterson, *Cryptography: Theory and Practice*, 4th ed., CRC Press, 2018.

Suggested Readings:

1. V. Nachef, J. Patarin, E. Volte, *Feistel Ciphers: Security Proofs and Cryptanalysis*, 1st ed., Springer, 2017.
2. M. Stamp and R. M. Low, *Applied Cryptanalysis: Breaking Ciphers in the Real World*, Wiley-IEEE Press, 2007.

AR/VR APPLICATION DEVELOPMENT (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
AR/VR Application Development	4	3	0	1	Foundations of Augmented and Virtual Reality

Course Objectives:
1. To understand the hardware and software architecture of AR/VR systems.
2. To design immersive user interfaces (UI) and user experiences (UX) tailored for spatial computing.
3. To develop interactive AR applications for mobile (ARCore/ARKit) and VR applications for headsets (Quest/Vive).
4. To explore the integration of spatial audio, haptics, and physics in virtual environments.

Course Outcomes:
By the end of this course, the student will be able to:
1. Identify and differentiate between various AR (Marker-based, Markerless) and VR (3DoF vs 6DoF) technologies.
2. Build functional 3D scenes with optimized assets and realistic lighting.
3. Implement complex interaction patterns such as teleportation, object manipulation, and gaze-based input.
4. Deploy cross-platform XR applications using OpenXR standards.

UNIT-I ***(12 hours)***
Introduction to XR Ecosystem: Taxonomy of Mixed Reality: The Milgram-Kishino Continuum; Difference between AR, VR, and MR;Hardware Foundations: Sensors (IMUs, Cameras, LIDAR), Displays (OLED, LCoS, Waveguides), and Tracking (Inside-out vs. Outside-in);Degrees of Freedom (DoF): 3DoF vs. 6DoF tracking; Human perception and the Vestibular system (reducing Motion Sickness);Development Tools: Overview of Unity, Unreal Engine, and WebXR.

UNIT-II ***(12 hours)***
Virtual Reality Development: Scene Construction: Importing 3D assets, Prefabs, and Scene hierarchy; VR Interaction Frameworks: XR Interaction Toolkit (XRI); Interactors, Interactables, and Sockets; Locomotion Systems: Teleportation, Continuous movement, and Snap-turn mechanics; Physics in VR: Rigidbodies, Colliders, and Trigger zones for immersive interaction.

UNIT-III ***(10 hours)***

Augmented Reality Development: AR Fundamentals: Environmental understanding (Plane detection), Motion tracking, and Light estimation.;Tracking Techniques: Image tracking (Marker-based) vs. Ground plane tracking (Markerless); Face tracking and Occlusion; AR Cloud & Persistence: Introduction to Spatial Anchors and Geo-spatial AR;UX for AR: Visual affordances, screen-space UI vs. world-space UI, and "The Hand" as an input device.

UNIT-IV ***(11 hours)***

Advanced XR Features and Optimization: Spatial Audio: 3D Sound spatialization, HRTF, and Doppler effect for immersion;Performance Optimization: Draw call batching, Occlusion culling, and LOD (Level of Detail) for mobile XR;Input Systems: Controller input, Hand tracking, Gaze-based selection, and Voice commands;Publishing: Build settings for Android/iOS (AR) and Standalone VR headsets (Meta Quest/PICO).

Practical Component: ***(30 hours)***

1. Environment Design: Create a 3D "Virtual Gallery" with baked lighting and optimized textures.
2. Basic Interaction: Script a "Physics Playground" where users can pick up, throw, and stack 3D objects using VR controllers.
3. Teleportation Logic: Implement a custom locomotion system with "Teleportation Areas" and "Teleportation Anchors."
4. Marker-based AR: Develop an AR mobile app that displays a 3D animated model when a specific "Image Target" (logo/business card) is scanned.
5. Furniture Placement (AR): Create a markerless AR app that allows users to place and rotate virtual furniture on a real-world detected floor.
6. Gaze-based Menu: Design a VR "Gaze UI" where looking at a button for 2 seconds triggers a scene change.
7. Hand Tracking Lab: Implement a "Virtual Piano" where users play notes using tracked finger movements (no controllers).

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. S. M. LaValle, *Virtual Reality*, Cambridge University Press, 2023.
2. J. Linowes, *Unity 2020 Virtual Reality Projects*, Packt Publishing, 2020.
3. D. Schmalstieg and T. Höllerer, *Augmented Reality: Principles and Practice*, Addison-Wesley / Pearson, 2016.

Suggested Readings:

1. J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality*, ACM Books / Morgan & Claypool, 2015.

2. Unity Technologies, *Unity Multiplayer Networking Documentation* (Netcode for GameObjects / Fusion), Unity Manual, latest edition.
3. M. Ball, *The Metaverse: And How It Will Revolutionize Everything*, Liveright Pub Corp, 2022

COMPUTER VISION FOR AR/VR (DSE-9/GE-9)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Computer Vision For AR/VR	4	3	0	1	Foundations of Augmented and Virtual Reality

Course Objectives:

1. Provide students with foundational knowledge of computer vision concepts and basic image processing techniques for AR/VR systems.
2. Introduce feature detection, description, and camera geometry techniques essential for tracking, pose estimation, and spatial understanding in AR/VR applications.
3. Enable students to analyze motion, tracking, and depth perception methods for immersive environments.
4. Introduce students to practical computer vision applications in AR/VR, including hand tracking, gesture recognition, and interactive scene understanding.
5. Provide students with the ability to understand vision-based AR/VR prototypes while considering real-time performance, latency, and optimization constraints.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand core computer vision principles essential to AR/VR systems.
2. Apply real-time vision algorithms for tracking, pose estimation, and environment mapping.
3. Integrate vision algorithms into AR/VR applications and evaluate system performance.
4. Analyze the limitations and challenges of vision-based perception

UNIT-I

Introduction to Computer Vision for AR/VR ***(12 hours)***

Definition and Scope of Computer Vision, Role of Vision in AR/VR Systems, Human Visual Perception and Its Relevance to AR/VR, Image Formation and Representation, Camera Models: Pinhole Camera Model, Perspective Projection, Basic Image Processing Operations: Filtering, Edge Detection, and Image Enhancement.

UNIT-II

Feature Detection and Camera Geometry ***(12 hours)***

Interest Point and Feature Detection Techniques, Feature Descriptors and Matching, Camera Calibration, Homogeneous Coordinates and 3D Transformations, Multiple View Geometry: Epipolar Geometry, Fundamental and Essential Matrices, Pose Estimation Techniques for AR Object Placement.

UNIT-III

Tracking, Motion Analysis, and Depth Perception ***(11 hours)***

Object Tracking Techniques - Marker-Based and Markerless Tracking, Optical Flow Methods - Lucas–Kanade and Horn–Schunck Algorithms, Motion Estimation and Visual Odometry, Depth Estimation Techniques: Stereo Vision, Disparity Maps, and Triangulation, Structure from Motion, Applications of Depth and Motion in AR/VR Interaction and Scene Understanding.

UNIT-IV

SLAM and AR/VR Applications ***(10 hours)***

Simultaneous Localization and Mapping, Visual SLAM and Visual-Inertial SLAM for AR/VR Devices, Sensor Fusion: Integration of Cameras, IMU, and Depth Sensors. Computer Vision in VR: Hand Tracking, Gesture Recognition, and Eye Tracking. Performance Considerations: Real-Time Constraints, Latency, and Optimization.

Practical Component: ***(30 hours)***

1. Study image representation and color space conversions using OpenCV.
2. Perform image preprocessing and enhancement using filtering and edge detection techniques.
3. Implement feature detection algorithms.
4. Perform feature matching and homography-based object recognition.
5. Develop a marker-based augmented reality system for virtual object overlay.
6. Implement optical flow algorithms for motion estimation in video sequences.
7. Design a real-time object tracking system using computer vision techniques.
8. Estimate depth using stereo vision and generate disparity maps.
9. Reconstruct 3D scenes using structure from motion techniques.
10. Implement a basic visual SLAM pipeline for localization and mapping.
11. Develop hand or gesture tracking for interactive AR/VR applications.
12. Integrate computer vision modules with Unity or Unreal Engine for AR/VR systems.
13. Analyze real-time performance, latency, and optimization in vision-based AR/VR applications.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. Richard Szeliski, Computer Vision: Algorithms and Applications.
2. Learning OpenCV 4 by Gary Bradski & Adrian Kaehler.
3. Programming Computer Vision with Python.

Suggested Readings:

1. Robotics Vision and Control by Peter Corke.
2. Dieter Schmalstieg and Tobias Höllerer, Augmented Reality: Principles and Practice.

FUNDAMENTALS OF MACHINE LEARNING (GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Fundamentals of Machine Learning	4	3	0	1	Fundamentals of Computer Programming, Fundamentals of Algorithms

Course Objectives:
1. To introduce the fundamental concepts and terminology of machine learning.
2. To understand different types of learning: supervised, unsupervised, and reinforcement learning.
3. To develop basic skills in data preprocessing, model building, and evaluation.
4. To apply machine learning techniques to real-world problems from diverse domains.
5. To provide hands-on experience using simple ML tools and Python-based libraries.

Course Outcomes:
At the end of this course, students will be able to:
1. Explain basic machine learning concepts and learning paradigms.
2. Prepare and preprocess datasets for analysis.
3. Build simple regression and classification models.
4. Evaluate model performance using standard metrics.
5. Apply machine learning techniques to domain-specific problems.

UNIT-I ***(10 hours)***
Introduction to Machine Learning and Data: Introduction to Machine Learning: definition, applications, and real-world use cases; Types of learning: supervised, unsupervised, and reinforcement learning;Overview of ML workflow; Types of data and data preprocessing: handling missing values, normalization, feature scaling; Introduction to Python and ML libraries (NumPy, Pandas, Scikit-learn).

UNIT-II ***(14 hours)***
Supervised Learning: Regression: Linear Regression, Multiple Regression; Classification: Logistic Regression, k-Nearest Neighbour (k-NN), Decision Trees; Model training and testing; Evaluation metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix, RMSE; Overfitting and underfitting.

UNIT-III ***(09 hours)***
Unsupervised Learning: Clustering: k-Means clustering, Hierarchical clustering; Dimensionality reduction: Introduction to PCA; Basic visualization techniques for high-dimensional data; Applications in business, healthcare, and social sciences.

UNIT-IV ***(12 hours)***

Introduction to Advanced Topics and Applications: Basics of Neural Networks, Introduction to model validation and cross-validation, Feature selection and importance, Ethical considerations in machine learning, Case studies from real world.

Practical Component: ***(30 hours)***

1. Data preprocessing and visualization using Python libraries.
2. Implementation of Linear Regression for real-world datasets.
3. Implementation of classification models (Logistic Regression, k-NN, Decision Tree).
4. Clustering using k-Means and visualization of clusters.
5. Dimensionality reduction using PCA.
6. Model evaluation using performance metrics and cross-validation.
7. Mini-project applying machine learning to a domain-specific dataset.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O'Reilly Media, 2022.
2. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.
3. S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed., Packt Publishing, 2019.

Suggested Readings:

1. T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
2. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed., Morgan Kaufmann, 2016.
3. J. D. Kelleher, *Deep Learning*, MIT Press, 2019.

FUNDAMENTALS OF CLOUD COMPUTING (GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Fundamentals of Cloud Computing	4	3	0	1	Foundations of Computer Networks

Course Objectives:

1. To introduce fundamental concepts and architecture of cloud computing.
2. To understand the service and deployment models of cloud platforms.
3. To explain virtualization, storage, and networking in cloud environments.
4. To provide basic hands-on experience in deploying applications in the cloud.
5. To develop an understanding of cloud security principles and governance.

Course Outcomes:

At the end of this course, students will be able to:

1. Explain cloud computing concepts, characteristics, and architecture.
2. Differentiate between cloud services and deployment models.
3. Describe virtualization, storage, and networking mechanisms in cloud systems.
4. Deploy and manage basic cloud-based applications.
5. Identify cloud security risks and apply fundamental protection mechanisms.

UNIT-I ***(11 hours)***

Introduction to Cloud Computing (Foundations): Evolution of Computing: Distributed Systems, Grid Computing, Cloud Definition and Essential Characteristics, Cloud Architecture: Front-end, Back-end, Middleware, Data Center Service Models: IaaS, PaaS, SaaS, Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Economics: Pay-as-you-go model, CAPEX vs OPEX, Cost-benefit considerations, Challenges: Vendor lock-in, Downtime, Compliance issues

UNIT-II ***(11 hours)***

Cloud Infrastructure and Virtualization: Virtualization Concepts: Hypervisors (Type I & Type II), Virtual Machines Containers (Basic Introduction), Resource Pooling and Multi-tenancy, Cloud Storage: Object Storage, Block Storage, File Storage, Introduction to Cloud Databases: Relational vs NoSQL (Conceptual level), Cloud Networking Basics: Virtual Private Cloud (VPC), Load Balancing, Content Delivery Network (CDN), Introduction to Cloud Service Providers

UNIT-III ***(11 hours)***

Cloud Application Deployment and Management: Cloud Application Architecture, Basic Cloud Design Principles, Introduction to DevOps in Cloud, CI/CD Overview (Conceptual), Basic Containerization using Docker, Introduction to Kubernetes, Serverless Computing (FaaS – Basic Concepts), Monitoring and Auto-scaling (Introduction).

UNIT-IV ***(12 hours)***

Cloud Security and Governance: Security Challenges in Cloud, Shared Responsibility Model, Identity and Access Management (IAM), Data Encryption (At Rest & In Transit), Backup and Disaster Recovery, Service Level Agreements (SLA), Compliance and Governance Basics, Ethical and Legal Issues in Cloud

Practical Component: ***(30 hours)***

1. Creating and configuring a Virtual Machine in a public cloud.
2. Deploying a static website using a cloud storage service.
3. Implementing basic Docker containerization.
4. Creating a simple serverless function.
5. Configuring IAM roles and policies.
6. Monitoring cloud resource usage and cost estimation.
7. Mini Project: Deploy a small web application on the cloud.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. T. Erl, R. Puttini, and Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, 1st ed., Prentice Hall, 2013.
2. Amazon Web Services, *AWS Documentation*, Available: https://docs.aws.amazon.com/
3. K. Hightower, B. Burns, and J. Beda, *Kubernetes: Up and Running: Dive into the Future of Infrastructure*, 2nd ed., O'Reilly Media, 2019.

Suggested Readings:

1. R. Buyya, C. Vecchiola, and S. Thamarai Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, 1st ed., Morgan Kaufmann, 2013.
2. T. Erl, R. Cope, and A. Naserpour, *Cloud Computing Design Patterns*, 1st ed., Pearson, 2015.

AUTOMATED MACHINE LEARNING (GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Automated Machine Learning	4	3	0	1	Fundamentals of Data Analytics

Course Objectives:

1. Introduce the principles and evolution of AutoML systems.
2. Reduce manual effort in ML pipeline design through automation.
3. Enable hands-on experience with state-of-the-art AutoML frameworks.
4. Familiarize students with optimization techniques used in AutoML.
5. Develop understanding of interpretability and trust in automatically generated models.
6. Prepare students for industrial-scale ML deployment and research.

Course Outcomes:

At the end of this course, students will be able to:

1. Understand the motivation, scope, and challenges of Automated Machine Learning.
2. Explain the components of an end-to-end AutoML pipeline.
3. Apply automated techniques for feature engineering, model selection, and hyperparameter optimization.
4. Implement AutoML frameworks for classification, regression, and time-series tasks.
5. Evaluate AutoML systems in terms of performance, efficiency, and interpretability.
6. Analyze ethical, reproducibility, and deployment issues related to AutoML systems.

UNIT-I ***(10 hours)***

Introduction to Automated Machine Learning: Motivation for AutoML; Challenges in Traditional ML Pipelines; End-to-End ML Workflow; Search Spaces and Optimization Objectives; Taxonomy of AutoML Systems; Automation vs Human-in-the-Loop; Overview of Industrial AutoML Platforms and Research Trends.

UNIT-II ***(10 hours)***

Automated Data Processing and Feature Engineering: Data Cleaning and Preprocessing Automation; Handling Missing Values and Outliers; Feature Encoding and Scaling; Automated Feature Extraction and Construction; Feature Selection Methods; Dimensionality Reduction; AutoML for Tabular, Text, and Image Data.

UNIT-III ***(15 hours)***

Model Selection and Hyperparameter Optimization: Automated Model Selection; Hyperparameter Optimization Techniques: Grid Search, Random Search, Bayesian Optimization; Multi-Fidelity Optimization; Early Stopping and Resource Allocation; Neural Architecture Search (NAS): Search Spaces, Search Strategies, and Performance Estimation.

UNIT-IV ***(10 hours)***

Advanced AutoML Systems and Deployment: AutoML Frameworks: Auto-sklearn, TPOT, H2O AutoML, AutoGluon; AutoML for Deep Learning; Explainability and Interpretability in AutoML; AutoML in Production: Model Monitoring and Drift Detection; Ethical Issues, Bias, and Reproducibility; Case Studies in Healthcare, Finance, and Industry.

Practical Component: ***(30 hours)***

1. Building a baseline ML pipeline manually and comparing it with AutoML results.
2. Using AutoML frameworks for classification and regression tasks.
3. Automating feature engineering and hyperparameter tuning.
4. Implementing Bayesian optimization for hyperparameter search.
5. Applying AutoML to time-series or real-world datasets.
6. Mini-project: Designing an end-to-end AutoML pipeline for a domain-specific problem.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. F. Hutter, L. Kotthoff, and J. Vanschoren (eds.), *Automated Machine Learning: Methods, Systems, Challenges*, 1st ed., Springer, 2019.
2. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed., MIT Press, 2016.
3. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, 1st ed., MIT Press, 2012.

Suggested Readings:

1. C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., Springer, 2006.
2. T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

FEDERATED LEARNING (DSE-6/GE-7)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Federated Learning	4	3	0	1	Fundamentals of Data Analytics

Course Outcomes:

At the end of this course, students will be able to:

1. Explain the FL setting and its relationship to distributed machine learning, including core actors, workflows, and design constraints.
2. Distinguish and model horizontal FL, vertical FL, and federated transfer learning scenarios and justify design choices for each.
3. Implement a horizontal FL (FedAvg-style) training workflow and evaluate convergence and performance under client heterogeneity.
4. Design and prototype vertical FL / federated transfer learning pipelines for feature-space and sample-space partitioned data.
5. Analyze incentive mechanism design goals in FL and implement a basic participation/reward scheme with measurable impact on training quality.
6. Build and report an FL application case-study (vision/language/recommendation or federated RL), including reproducible experimentation and a data-protection checklist.

Course Objectives:

1. Introduce federated learning (FL) as a privacy-aware collaborative learning paradigm and position it relative to distributed machine learning.
2. Develop the ability to classify FL problem settings as horizontal FL, vertical FL, or federated transfer learning and choose an appropriate workflow.
3. Enable students to implement core FL training loops and aggregation workflows for real datasets under practical constraints.
4. Train learners to reason about confidentiality, privacy, and data-protection drivers for FL, including legal considerations (e.g., GDPR-style requirements).
5. Develop an understanding of incentive mechanism design for federated learning participation and its impact on system performance.
6. Enable students to apply FL patterns to representative application families (vision, language, recommendation, and federated RL).

UNIT-I ***(10 hours)***

Introduction, Background, and Distributed Machine Learning Foundations for FL: motivation for federated learning and data confidentiality, privacy and data-protection drivers and constraints, background concepts and terminology for federated machine learning, distributed machine learning fundamentals relevant to FL (data locality, coordination, aggregation), high-level privacy-preserving machine learning idea space and why FL differs from centralized training, end-to-end FL workflow overview and design constraints, overview of representative use-case motivations and responsible deployment considerations.

UNIT-II ***(12 hours)***

Horizontal Federated Learning: horizontal federated learning (HFL) setting and assumptions, client-server orchestration and round-based training, local training and global aggregation intuition, model update aggregation patterns and practical tuning levers, handling client participation variability and practical system considerations consistent with HFL, evaluation of federated models in HFL settings, summary of canonical HFL method motivations (including iterative model averaging as a representative approach).

UNIT-III ***(12 hours)***

Vertical Federated Learning and Federated Transfer Learning: vertical federated learning (VFL) setting with feature-space partitioned data and overlapping samples, coordination patterns for VFL workflows, role of privacy-preserving computation concepts at a high level as used to enable collaboration without raw data sharing, federated transfer learning (FTL) motivation for limited sample overlap and heterogeneous feature spaces, conceptual workflows for aligning knowledge across parties in FTL settings, comparative discussion of HFL vs VFL vs FTL assumptions and practical selection guidance.

UNIT-IV ***(11 hours)***

Incentive Mechanisms, Application Families, and Legal/Data-Protection Context: incentive mechanism design motivations in federated learning and how incentives affect participation and learning outcomes, high-level economic/game-theoretic framing for incentivizing collaboration, federated learning patterns for computer vision, language, and recommendation settings, introduction to federated reinforcement learning as an FL application family, selected application case patterns and deployment considerations, summary and outlook themes including open challenges, legal development on data protection with emphasis on GDPR-style principles that motivate privacy-aware ML workflows.

Practical Component: ***(30 hours)***

1. Setup and reproducibility: Python environment, experiment folder structure, logging, seeding, and a simple client–server simulation harness.
2. Distributed ML warm-up: implement centralized training vs distributed data-local training baseline to observe differences in data movement and orchestration costs.
3. Horizontal partitioning lab: create synthetic HFL client datasets (IID vs non-IID splits) and quantify heterogeneity (class distribution, sample counts).
4. Implement an HFL training loop: local training + server aggregation (FedAvg-style), run multi-round training, and plot performance across rounds.
5. Participation variability lab: simulate partial participation and dropped clients; study the impact on convergence and final accuracy.
6. Communication accounting: measure bytes/round and wall-clock time for HFL runs; compare at least two settings of local epochs and client fractions.
7. Vertical FL data modeling: construct a toy VFL dataset (two parties with different feature subsets and overlapping sample IDs) and define a collaborative objective.
8. Prototype a VFL workflow: implement a minimal secure "no-raw-feature-sharing" pipeline using feature partitioning and controlled intermediate exchanges (toy protocol sufficient for learning demonstration).
9. Federated transfer learning exercise: design a small FTL-inspired setup with limited overlap and heterogeneous feature spaces, implement a baseline transfer strategy, and document assumptions and limitations.

10. Incentive mechanism simulation: define a simple participation utility and a reward rule, simulate different incentive settings, and report the effect on participation rates and training quality.
11. Application case-study: choose one application family (vision or language or recommendation or federated RL) and implement a small-scale federated prototype with reproducible reporting.
12. Data-protection checklist: write a short compliance-oriented checklist for the mini-project (data minimization rationale, access controls, logging policy, retention, and GDPR-style considerations) and attach it to the final report.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, Federated Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool / Springer, 2019.
2. P. Kairouz, H. B. McMahan, et al., Advances and Open Problems in Federated Learning, Foundations and Trends in Machine Learning, Now Publishers / Emerald Publishing, 2021.

Suggested Readings:

1. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data: General Data Protection Regulation (GDPR), 2016.
2. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.
3. K. Bonawitz, V. Ivanov, B. Kreuter, et al., “Practical Secure Aggregation for Federated Learning on User-Held Data,” NIPS 2016 workshop on Private Multi-Party Machine Learning, 2016.

APPLIED GENERATIVE AI FOR DATA SCIENCE (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Applied Generative AI for Data Science	4	3	0	1	Fundamental of Data Analytics

Course Outcomes:
At the end of the course, students will be able to

1. Understand the fundamental concepts, architectures, and working principles of Generative Artificial Intelligence models.
2. Understand and apply generative AI techniques for the creation of images and digital paintings.
3. Understand the methods and models used for AI-based music and audio generation.
4. Understand the ethical, legal, and societal implications of Generative AI in creative and interactive applications.

Course Objectives:
- To develop a strong conceptual understanding of the fundamental principles, architectures, and learning paradigms underlying Generative Artificial Intelligence.
- To familiarize students with AI-driven techniques for the generation of images, artistic paintings, music, and interactive play environments.
- To provide practical exposure to state-of-the-art generative models and tools, enabling students to design, implement, and evaluate AI-driven content generation systems.

UNIT-I ***(11 hours)***
Introduction to Generative Artificial Intelligence (Gen AI): Introduction to Gen AI, Evolution and Historical Overview of Generative modeling, Difference between Gen AI and Discriminative Modeling, Capabilities of Generative AI and its Use Cases in Real World, Applications of Generative AI in Different Sectors: Healthcare, Finance, Marketing, Media, and Education, Role of Gen AI Models and Tools for Text, Code, Image, Audio, and Video Generation; Types of Generative Models- GANs, VAEs, Autoregressive Models, and Vector Quantized Diffusion Models.

UNIT-II ***(10 hours)***
Generative Models for Text: Language Model Basics, Building Blocks of Language Models, Transformer Architecture, Encoder and Decoder, Attention Mechanisms, Generation of Text, Models like BERT and GPT Models, Generation of Text, Autoencoding, Regression Models, Exploring ChatGPT, Prompt Engineering, Design Prompts, Revising Prompts using Reinforcement Learning, Retrieval Augmentation Generation, Multimodal LLM, and Issues of LLM.

UNIT-III ***(11 hours)***
Generation of Images: Generative Adversarial Networks, Adversarial Training Process, Nash Equilibrium, Variational Autoencoders, Stable Diffusion Models, Introduction to Transformer based Image Generation, CLIP, Visual Transformers ViT, Dall-E2, and Dall-E3. Generation of Painting, Music

and Play- Variants of GANs, Types of GANs, Cyclic GAN, Using Cyclic GAN to Generate Painting, Music Generating RNN, MuseGAN, Autonomous Agents- Deep Q Network, Actor-Critic Network.

UNIT-IV ***(10 hours)***

Gen AI Tools and Applications: Overview of GPT-based and AI-driven tools for data querying and preparation. Application of Generative AI in data cleaning, feature engineering, and exploratory analysis. Real-world use cases of Gen AI in enhancing data science workflows. Integration of Generative AI techniques within machine learning model development and deployment.

Practical Component: ***(30 hours)***

1. Implementation of autoencoders and variational autoencoders for basic generative modeling.
2. Fine-tuning and experimentation with Transformer-based language models for text generation and prompt engineering.
3. Implementation of GAN and CycleGAN models for image synthesis and style transfer.
4. Experimentation with Stable Diffusion and multimodal models such as CLIP/ViT for text-to-image tasks.
5. Music or sequential data generation using RNN/LSTM-based architectures.
6. Application of Generative AI tools in data science workflows including automated querying, feature engineering, and report generation.
7. Mini-project involving design and implementation of a Generative AI application with experimental evaluation and ethical assessment.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. A. Vemula, *Practical Generative AI for Data Science: From Theory to Real-World Applications*, 2024.
2. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
3. A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O'Reilly Media, 2022.
4. D. Foster, *Generative Deep Learning*, 2nd ed., O'Reilly Media, 2022.

Suggested Readings:

1. L. Raschka, *Build a Large Language Model (From Scratch)*, Manning Publications, 2024.
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, pp.1877-1901.
3. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763).

DATA PRIVACY, SECURITY & REGULATORY COMPLIANCE (DSE-10/GE-10)

CREDIT DISTRIBUTION, ELIGIBILITY, AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Data Privacy, Security & Regulatory Compliance	4	3	0	1	Fundamentals of Data Analytics

Course Objectives:
1. To understand the foundational concepts of data privacy and security.
2. To explore privacy-preserving techniques in data science and machine learning.
3. To examine regulatory frameworks such as GDPR, HIPAA, and CCPA.
4. To develop skills to design and implement privacy-compliant data systems.

Course Outcomes:
At the end of the course, students will be able to:
1. Understand core principles of data privacy, security, and privacy-preserving techniques.
2. Analyze privacy attacks and design effective security mitigations.
3. Apply privacy-aware techniques in machine learning and distributed data systems.
4. Interpret and implement regulatory compliance requirements in data science projects.

UNIT-I ***(12 hours)***
Foundations of Data Privacy and Security: Introduction to Data Privacy, Data Protection, and Information Security, Privacy vs. Security: Concepts, Scope and Differences, Role of Data Privacy in Data Science and Analytics, Threats to Data Privacy in Modern Data-Driven Systems, Basic Cryptographic Concepts for Data Protection, Access Control Mechanisms: Authentication and Authorization, Data Anonymization, K-Anonymity and its Limitations, Differential Privacy: Definition and Motivation, Noise-based Privacy Mechanisms: Gaussian Noise Mechanism and Laplace Mechanism, Design Appropriate Privacy Measures, Differential Privacy Libraries in Pipelines.

UNIT-II ***(12 hours)***
Privacy Attacks, Threats, and Security Mitigation: Privacy Attacks: Analyzing Common Attack Vectors, Netflix Price Attack, Linkage Attack, Singling Out Attack, Strava Heap Map Attack, Membership Inference Attack, Inferring Sensitive Attributes, Attacks Against Privacy Protocols, Data Security in Privacy Protection, Data Loss Prevention Techniques, Additional Security Controls for Sensitive Data, Threat Modeling for Privacy and Security, Incident Response and Breach Management, Data Security Mitigations: Web Security Basics for Data Applications, Protecting Training Data, ML Models, and Inference APIs.

UNIT-III ***(11 hours)***
Privacy-Aware Machine Learning and Data Science: Privacy Challenges in ML, Privacy-Preserving Techniques in Data Science or ML, Differentially Private Stochastic Gradient Descent, Open-source Libraries for Privacy-Preserving ML, Designing Privacy-by-design ML Systems, Distributed Data and Privacy Challenges, Motivation for Distributed and Decentralized Data Analytics, Differential Privacy in Distributed Data, Federated Learning (FL): A Brief History, Why, When, and How to use FL, Federated

Learning Architecture and Workflows, Security and Privacy Threats in FL, Deploying FL Frameworks, Open Source Federated Libraries: Flower.

UNIT-IV ***(10 hours)***

Navigating the Legal Side of Privacy: Introduction to Legal and Regulatory Aspects of Data Privacy, General Data Protection Regulation (GDPR): Scope and Objectives, Fundamental Rights Under GDPR, Privacy-Enhancing Technologies for GDPR, The GDPR's Data Protection Impact Assessment: Agile and Iterative Risk Assessment, GDPR Compliance Challenges in Data Science and AI Systems, California Consumer Privacy Act (CCPA), Other Global Regulations: HIPPA, LGPD, and PIPL, Reading Privacy Policies and Terms of Service, Reading Data Processing Agreements, Data Governance Principles and Frameworks, Data Governance 2.0: Federated Governance, Governance challenges in anonymized and privacy-aware ML systems.

Practical Component: ***(30 hours)***

1. Implementation of basic cryptographic techniques, access control mechanisms, and anonymization methods (k-anonymity) on structured datasets.
2. Implementation of Differential Privacy mechanisms (Laplace and Gaussian noise) using open-source libraries and integration into ML pipelines.
3. Simulation and analysis of privacy attacks (linkage attack, membership inference attack) and design of mitigation strategies.
4. Implementation of privacy-preserving machine learning techniques including DP-SGD and experimentation with federated learning using open-source frameworks (e.g., Flower).
5. Threat modeling, data loss prevention strategies, and security configuration for ML model deployment and inference APIs.
6. Case study based compliance assessment project involving GDPR/CCPA/HIPAA analysis, DPIA preparation, and privacy-by-design system documentation.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. K. Jarmul, *Enhancing Privacy and Security in Data*, Foreword by N. D. Stefflbauer, 1st ed., O'Reilly Media, 2023.
2. P. Voigt and A. von dem Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*, 1st ed., Springer, 2017.
3. C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, Foundations and Trends® in Theoretical Computer Science, Now Publishers, 2014.

Suggested Readings:

1. C. Kuner, L. A. Bygrave, and C. Docksey (eds.), *The EU General Data Protection Regulation (GDPR): A Commentary*, 1st ed., Oxford University Press, 2020.
2. A. Heather, B. Betsy, B. Paul, and L. Piotr, O. Ana, and S. Adam, *Building Secure and Reliable Systems*, 1st ed., O'Reilly Media, 2020.

SECURE SOFTWARE ENGINEERING (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Secure Software Engineering	4	3	0	1	Software Engineering, Object Oriented Software Engineering

Course Objectives:

1. To introduce the fundamental principles of software security, secure software development lifecycle, and threat modeling techniques.
2. To explain secure design principles, trust boundary enforcement, and identity and access management mechanisms for building secure software systems.
3. To study common classes of software vulnerabilities and the techniques used for their detection, analysis, and mitigation.
4. To expose students to secure software testing, deployment, and operational security practices, including DevSecOps, container security, and incident response.
5. To develop the ability to analyze, design, and evaluate secure software systems using industry standards, tools, and real-world case studies.

Course Outcomes:

After successful completion of the course, students will be able to:

1. Explain fundamental security principles, threat models, and risk assessment techniques used in secure software development.
2. Apply secure design principles and trust boundary enforcement techniques to design resilient software architectures.
3. Identify, analyze, and mitigate common software vulnerabilities using secure coding, review, and testing practices.
4. Use industry-standard open-source security tools to perform static, dynamic, and dependency security analysis.
5. Integrate security practices into software deployment and operations, including CI/CD pipelines, container platforms, and incident response.

UNIT-I ***(10 hours)***

Foundations of Security & Threat Modeling: Security principles: confidentiality, integrity, availability, and accountability; Secure Software Development Lifecycle (SSDLC) and integration with Agile/Waterfall models; Threat modeling concepts: assets, adversaries, attack surfaces, STRIDE, and attack trees; Risk assessment using DREAD/CVSS; Overview of security standards and compliance (OWASP, NIST, ISO/IEC 27001).

UNIT-II ***(13 hours)***

Secure Design Principles: Saltzer and Schroeder's secure design principles; Trust boundaries and secure data flow using DFDs; Input validation, canonicalization, output encoding, and taint tracking; Positive security models and policy enforcement; Identity and Access Management (IAM): authentication (MFA, OAuth2, OIDC) and authorization models (RBAC, ABAC).

UNIT-III ***(12 hours)***

Software Vulnerabilities and Security Testing: Common vulnerabilities: buffer overflows, injection attacks, race conditions, and dependency risks; Memory safety and secure coding practices; Static (SAST), dynamic (DAST), and software composition analysis (SCA); Secure code review techniques and fuzz testing.

UNIT-IV ***(10 hours)***

Secure Deployment and Operations: Security integration in CI/CD pipelines (Shift Left approach) Secrets management and secure configuration practices; Container and cloud security (Docker, Kubernetes basics); Logging, monitoring, incident response, and forensic readiness.

Practical Component: ***(30 hours)***

1. Threat modeling and secure design exercises using standard security frameworks
2. Static and dynamic security testing using open-source SAST and DAST tools
3. Dependency and vulnerability analysis of third-party libraries
4. Secure code review and automated fuzz testing exercises
5. Security integration in CI/CD pipelines using automated security tools

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. M. Graff and K. R. van Wyk, *Secure Coding: Principles and Practices*, 1st ed., O'Reilly Media, 2003.
2. M. Dowd, J. McDonald, and J. Schuh, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*, 1st ed., Addison-Wesley, 2006.
3. A. Shostack, *Threat Modeling: Designing for Security*, 1st ed., Wiley, 2014.
4. T. Janca, *Alice and Bob Learn Application Security*, 1st ed., Wiley, 2020.

Suggested Readings:

1. R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed., Wiley, 2008.
2. R. C. Seacord, *Secure Coding in C and C++*, 2nd ed., Addison-Wesley, 2013.
3. OWASP Foundation, *OWASP Top 10: The Ten Most Critical Web Application Security Risks*, Latest ed., OWASP Foundation.
4. OWASP Foundation, *Application Security Verification Standard (ASVS)*, Latest ed., OWASP Foundation.

5. International Organization for Standardization (ISO), *ISO/IEC 27001: Information Security Management Systems — Requirements*, Latest ed., ISO.
6. International Organization for Standardization (ISO), *ISO/IEC 27002: Information Security Controls*, Latest ed., ISO.

BUILD, RELEASE AND DEPLOYMENT ENGINEERING (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Build, Release and Deployment Engineering	4	3	0	1	Object Oriented Software Engineering

Course Objectives:

1. To introduce modern software delivery models and build–release–deployment engineering practices.
2. To develop understanding of configuration management, version control, semantic versioning, and continuous integration.
3. To enable design and implementation of automated build, test, and deployment pipelines.
4. To familiarize students with continuous delivery ecosystems including infrastructure, environments, and dependency management.
5. To expose students to DevOps and DevSecOps practices including containerization, orchestration, Infrastructure as Code, cloud deployment, monitoring, and security integration

Course Outcomes:

On successful completion of the course, students will be able to:

1. Explain modern software delivery models and build–release–deployment engineering practices.
2. Apply configuration management, version control, and semantic versioning in software projects.
3. Design and implement CI/CD pipelines with automated builds, testing, and quality assurance.
4. Develop and deploy containerized applications using orchestration platforms and cloud environments.
5. Apply Infrastructure as Code principles to provision and manage scalable infrastructure.
6. Integrate DevOps and DevSecOps practices including monitoring, security controls, and vulnerability management within automated pipelines.

UNIT-I ***(11 hours)***

Foundations and Continuous Integration: Evolution of software delivery. Role of Build, Release, and Deployment Engineering in Software Lifecycle. Configuration Management. Version control systems. Semantic versioning and release numbering schemes. Continuous Integration (CI): concepts and benefits. CI pipeline stages. CI principles. CI with distributed teams. Automated testing in CI pipelines. Implementing a Testing strategy.

UNIT-II ***(12 hours)***

Deployment Pipeline: Deployment practices. Preparing to release. Implementing a deployment pipeline. Build tools. Principles and Practices of Build and Deployment Scripting. The Commit Stage. Principles and Practices. Automated Acceptance Testing. Testing Nonfunctional requirements. Deploying and Releasing applications

UNIT-III ***(8 hours)***

Delivery Ecosystem: Managing Infrastructure and Environments. Managing Data. Managing Components and Dependencies. Managing Continuous Delivery.

UNIT-IV ***(14 hours)***

Modern DevOps Practices and Scalable Deployment: Containerization and Orchestration. Introduction to Docker and containerization. Building and managing Docker containers. Introduction to Kubernetes. Kubernetes architecture and deployment strategies. Infrastructure as Code (IaC). Design Patterns for Infrastructure. Scaling IaC with Teams. DevOps in the cloud (AWS, Azure, GCP). DevSecOps and security best practices. Secrets management (Vault, AWS Secrets Manager). Vulnerability scanning (Trivy, SonarQube)

Practical Component: ***(30 hours)***

1. Implementation of version control workflows, branching strategies, and configuration management practices using distributed VCS.
2. Design and implementation of Continuous Integration pipelines with automated build, unit testing, and quality checks.
3. Construction of deployment pipelines including commit stage, automated acceptance testing, and release automation.
4. Management of infrastructure, environments, data, and dependencies in a continuous delivery ecosystem.
5. Development and deployment of containerized applications using Docker and orchestration with Kubernetes.
6. Implementation of Infrastructure as Code and DevSecOps practices including cloud provisioning, secrets management, and vulnerability scanning.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, 1st ed., Addison-Wesley, 2010.
2. K. Morris, *Infrastructure as Code: Managing Servers in the Cloud*, 1st ed., O’Reilly Media, 2016.
3. N. Turnbull, *The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise*, 1st ed., IT Revolution Press, 2019.
4. G. Kim, J. Humble, P. Debois, and J. Willis (eds.), *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, 2nd ed., IT Revolution Press, 2021.

Suggested Readings:

1. R. Wang, *Infrastructure as Code, Patterns and Practices: With Examples in Python and Terraform*, 1st ed., Manning Publications, 2022.
2. B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, *Kubernetes: Up and Running*, 3rd ed., O'Reilly Media, 2022.

AI IN CYBERSECURITY (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
AI in Cybersecurity	4	3	0	1	Cryptography Essentials

Course Outcomes:
At the end of this course, students will be able to:
1. Understand key AI and machine learning concepts relevant to cybersecurity applications.
2. Analyze cybersecurity threats using supervised and unsupervised learning techniques.
3. Design and implement AI-driven security solutions such as intrusion detection, malware classification, and behavioral profiling.
4. Evaluate the performance of AI models in real-world cybersecurity scenarios.
5. Identify ethical, legal, and privacy concerns when deploying AI in cybersecurity systems.

Course Objectives:
1. Introduce fundamental AI and machine learning concepts as applied to cybersecurity.
2. Understand how AI techniques enhance cybersecurity practices such as intrusion detection and malware detection.
3. Explore anomaly detection, behavioral analysis, and intelligent threat hunting using AI.
4. Familiarize students with practical tools and frameworks for implementing AI-based security solutions.
5. Analyze ethical, legal, and safety considerations in AI-enabled cybersecurity systems.

UNIT-I ***(11 hours)***
Introduction to AI and Cybersecurity: Introduction to cybersecurity and AI; Cyber threat landscape and attack vectors; Fundamentals of AI, machine learning (supervised and unsupervised), and their role in cybersecurity; Python programming environment setup for AI in security; Data collection and preprocessing for security datasets.

UNIT-II ***(11 hours)***
AI Techniques for Security Analytics: Supervised learning for intrusion detection and classification (decision trees, SVMs, neural networks); Unsupervised learning for anomaly and behavioral analysis; Feature extraction and selection for security data; Time-series analysis for network traffic behaviors; Evaluation metrics for classifier performance (precision, recall, ROC).

UNIT-III ***(11 hours)***
AI for Malware and Threat Detection: AI-driven malware analysis and classification; Deep learning approaches for malware detection; Natural language processing (NLP) in phishing and spam detection; Botnet detection and profiling; Adversarial attacks on ML models and defenses; Generative models and security applications.

UNIT-IV ***(12 hours)***

Practical AI Security Systems and Ethics: AI for Security Information and Event Management (SIEM); Automated incident response with AI; Explainable AI (XAI) for cybersecurity decision support; Ethical, legal, and privacy considerations in AI use for security; Emerging topics: autonomous security systems, reinforcement learning in cyber defense, future trends.

Practical Component: ***(30 hours)***

1. Setting up AI development environments (Python, TensorFlow/PyTorch, Scikit-Learn).
2. Data preprocessing and feature engineering for network security datasets.
3. Implementing supervised classifiers for intrusion detection.
4. Unsupervised anomaly detection for network traffic.
5. Deep learning-based malware classification.
6. Use of NLP for phishing and spam detection.
7. Security evaluation of AI models under adversarial conditions.
8. Case studies and mini-projects on building AI-enhanced cybersecurity applications.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. C. Chio and D. Freeman, *Machine Learning and Security: Protecting Systems with Data and Algorithms*, 1st ed., O'Reilly Media, 2018.
2. A. Parisi, *Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies*, pPackt Publishing Limited, 2019.
3. S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*, CRC Press, 2011.

Suggested Readings:

1. O. Santos, S. Salam, and H. Dahi, *AI Revolution in Networking, Cybersecurity, and Emerging Technologies*, 1st ed.,Addison-Wesley Professional, 2024.
2. S. Mahajan, M. Khurana, and V. V. Estrela, *Applying Artificial Intelligence in Cybersecurity Analytics and Cyber Threat Detection*, 1st ed. Wiley, 2024.

QUANTUM CRYPTOGRAPHY (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Quantum Cryptography	4	3	0	1	Cryptography Essentials

Course Outcomes:

At the end of this course, students will be able to:

1. Understand quantum bits (qubits) and quantum state representations.
2. Explain the principles of quantum key distribution (QKD) protocols and security.
3. Analyze the role of entanglement and uncertainty in cryptographic settings.
4. Evaluate advanced quantum cryptographic primitives beyond QKD.
5. Understand challenges, device independence, and future directions in quantum cryptography.

Course Objectives:

1. To introduce the fundamentals of quantum mechanics as they apply to information processing.
2. To study quantum communication concepts and how they differ from classical cryptography.
3. To explore quantum key distribution and security proofs in quantum settings.
4. To understand advanced quantum cryptographic protocols and their security guarantees.

UNIT-I ***(11 hours)***

Introduction to Quantum Information and Cryptography: Fundamentals of quantum mechanics for information theory: qubits, superposition, measurement, density matrices, partial trace; Basics of quantum states and quantum measurements; Classical vs. quantum information; Overview of quantum cryptography and its significance; Comparison with classical cryptography.

UNIT-II ***(11 hours)***

Quantum Key Distribution (QKD): Security of classical key distribution vs. quantum key distribution; BB84 protocol; Information reconciliation and privacy amplification; Entanglement-based QKD protocols; Device-independent QKD and security from uncertainty relations; Practical considerations and imperfections in QKD systems.

UNIT-III ***(11 hours)***

Entanglement and Advanced Quantum Protocols: Quantum entanglement and its cryptographic uses; Purifications and monogamy of entanglement; Secret sharing via quantum states; Quantum bit commitment (limitations and impossibility results); Quantum coin flipping; Oblivious transfer and position verification

UNIT-IV ***(12 hours)***

Security Analysis and Emerging Topics: Security proofs in quantum cryptography: trace distance, fidelity, von Neumann and min-entropy, uncertainty principle as a guessing game; Quantum attacks on classical cryptography (Shor's and Grover's algorithms overview); Post-quantum cryptography and quantum-resistant primitives; Future directions: quantum internet and network standards for QKD.

Practical Component ***(30 hours)***

1. Simulation and implementation of BB84 QKD protocol using quantum computing simulators (e.g., Qiskit).
2. Experiments with quantum random number generation.
3. Quantum measurement and state preparation tasks.
4. Security analysis of simple QKD implementations under noise and eavesdropping models.
5. Simulation of basic entanglement and purification processes.
6. Comparison of classical vs. quantum key distribution performance

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary ed., Cambridge University Press, 2010.
2. G. Van Assche, *Quantum Cryptography and Secret-Key Distillation*, 1st ed., Cambridge University Press, 2012.
3. M. M. Wilde, *Quantum Information Theory*, 2nd ed., Cambridge University Press, 2017.

Suggested Readings:

1. P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2006.
2. T. Vidick, S. Wehner, *Introduction to Quantum Cryptography*, Cambridge University Press, 2023.

AR/VR HARDWARE SYSTEMS: CONCEPTS AND ARCHITECTURES (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
AR/VR Hardware Systems: Concepts and Architectures	4	3	0	1	Foundations of Augmented and Virtual Reality

Course Objectives:
1. To introduce students to the fundamental hardware components that enable immersive AR/VR experiences.
2. To provide an understanding of display technologies, sensing modalities, and processing architectures used in modern AR/VR devices.
3. To familiarize students with system-level challenges such as latency, power consumption, heat dissipation, and ergonomics.
4. To enable students to analyze commercial AR/VR platforms from a hardware architecture perspective.
5. To equip students with the skills required to interface sensors, displays, and compute units in AR/VR prototypes.

Course Outcomes:
At the end of this course, students will be able to:
1. Understand the hardware building blocks of AR/VR systems, including display, sensing, and processing units.
2. Analyze AR/VR device architectures with respect to performance, latency, power, and user comfort.
3. Evaluate trade-offs in AR/VR hardware design for immersive and real-time applications.
4. Design and prototype basic AR/VR hardware subsystems considering real-world constraints.

UNIT-I
Introduction to AR/VR Hardware Systems ***(12 hours)***
Overview of Augmented Reality and Virtual Reality Hardware; Evolution of AR/VR Devices; System-Level Architecture of AR/VR Platforms; Hardware vs Software Responsibilities in AR/VR Systems; Key Performance Metrics: Latency, Frame Rate, Field of View (FoV), Resolution, and Refresh Rate; Hardware Challenges in Immersive Computing.

UNIT-II
Display Technologies for AR/VR ***(13 hours)***
Human Visual System and Display Requirements; Near-Eye Display Systems; Optical See-Through vs Video See-Through Displays; Head-Mounted Displays; Various Display Technologies; Optical

Components: Lenses, Mirrors, and Combiners; Vergence–Accommodation Conflict and Mitigation Techniques.

UNIT-III
Sensing, Tracking, and Input Hardware ***(10 hours)***
Tracking Requirements in AR/VR Systems; Positional and Rotational Tracking; Inside-Out vs Outside-In Tracking; Sensors for AR/VR; Hand and Gesture Tracking Hardware; Sensor Calibration and Synchronization; Sensor Fusion at Hardware Level.

UNIT-IV
Processing Architectures and System Integration ***(10 hours)***
Processing Units for AR/VR; Edge vs On-Device Processing; Memory Systems and Bandwidth Requirements; Power Management and Thermal Design; Hardware Latency Pipeline and Optimization; Future Trends in AR/VR Hardware.

Practical Component: ***(30 hours)***
1. Study of internal hardware architecture of commercial AR/VR headsets.
2. Interfacing IMU sensors with microcontrollers for orientation tracking.
3. Camera-based positional tracking using embedded platforms.
4. Analysis of display parameters such as refresh rate, resolution, and latency.
5. Sensor fusion implementation using IMU and camera data.
6. Power consumption analysis of AR/VR hardware components.
7. Hands-on experimentation with AR/VR development kits (e.g., Oculus, HoloLens, mobile-based AR).
8. Hardware latency measurement and performance evaluation.

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:
1. Virtual Reality by Steven M. LaValle
2. Understanding Virtual Reality by William R. Sherman & Alan B. Craig
3. Augmented Reality: Principles and Practices by Dieter Schmalstieg & Tobias Hollerer

Suggested Readings:
1. S. Aukstakalnis, Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR.
2. J. Jerald, The VR Book: Human-Centered Design for Virtual Reality, ACM.
3. B. Furht, Handbook of Augmented Reality, Springer, 2011.

ETHICS, PRIVACY, AND SAFETY IN XR SYSTEMS (DSE-10/GE-10)

CREDIT DISTRIBUTION AND PREREQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Ethics, Privacy, and Safety in XR Systems	4	3	0	1	Foundations of Augmented and Virtual Reality

Course Objectives:

1. To introduce ethical challenges in XR (AR/VR/MR) systems.
2. To understand privacy risks arising from immersive technologies.
3. To study psychological, social, and physical safety concerns in XR environments.
4. To analyze regulatory, legal, and governance frameworks for XR technologies.
5. To develop responsible XR design practices aligned with industry standards.

Course Outcomes:

At the end of this course, students will be able to:

1. Analyze ethical implications of immersive XR systems.
2. Evaluate privacy risks related to biometric, behavioral, and spatial data.
3. Identify psychological, physical, and social safety threats in XR platforms.
4. Apply privacy-by-design and safety-by-design principles in XR applications.
5. Design XR systems aligned with regulatory and responsible innovation frameworks.

UNIT-I ***(11 hours)***

Foundations of Ethics in XR Systems: Introduction to immersive technologies and societal impact; Ethical frameworks: consequentialism, deontology, virtue ethics in technology design; Digital embodiment, identity, and behavioral manipulation in XR; Algorithmic bias, accessibility, and inclusivity in immersive systems; Case studies: persuasive design, dark patterns, and behavioral influence in VR.

UNIT-II ***(11 hours)***

Privacy in XR Systems: Data types in XR: biometric data, gaze tracking, motion data, spatial mapping; Privacy risks in immersive environments: inference attacks, behavioral profiling; Surveillance, bystander privacy, and spatial data governance; Privacy-by-design principles in AR/VR development; Legal and regulatory frameworks: GDPR principles, data minimization, consent in immersive systems.

UNIT-III ***(11 hours)***

Safety and Well-being in Immersive Environments: Physical safety: motion sickness, spatial disorientation, and ergonomic risks; Psychological effects: addiction, desensitization, identity dissociation, harassment in virtual spaces; Social safety: toxicity, cyberbullying, and avatar-based abuse; Content moderation and governance in multi-user XR platforms; Ethical challenges in AI-driven immersive agents and virtual influencers.

UNIT-IV ***(12 hours)***

Responsible XR Design and Governance: Safety-by-design and ethical design methodologies; Risk assessment frameworks for XR systems; Transparency, explainability, and accountability in immersive AI systems; Industry standards and policy guidelines for XR platforms; Future challenges: metaverse governance, digital twins, biometric security, and neuro-rights.

Practical Component: ***(30 hours)***

1. Ethical analysis of a popular AR/VR application using a simple ethical framework template.
2. Identification of privacy risks in XR applications using data flow diagrams.
3. Basic design of a privacy-by-design feature (e.g., consent mechanism, anonymization).
4. Safety evaluation of XR platforms using a checklist-based approach.
5. Case study discussion on real-world XR incidents (privacy/safety issues).
6. Group activity: design guidelines for responsible XR systems.
7. Mini-project: Design a conceptual XR application focusing on Ethical considerations, Privacy safeguards and User safety features

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Essential Readings:

1. E. Langran and M. Slater, *Virtual Reality and Ethics: Ethical Issues in Immersive Technologies*, Routledge, 2020.
2. E. J. Ramirez, *The Ethics of Virtual and Augmented Reality: Building Worlds*, 1st Ed., Routledge, 2024.
3. M. Cotton, *Virtual Reality, Empathy and Ethics*, Springer (Palgrave Macmillan), 2021.
4. B. Friedman and D. G. Hendry, *Value Sensitive Design: Shaping Technology with Moral Imagination*, MIT Press, 2019.
5. EU GDPR Regulation (General Data Protection Regulation) – Selected Articles relevant to immersive systems.

Suggested Readings:

1. M. J. Grabowski, *Ethics of Virtual Reality: Technology and Experience*, Lexington Books / Bloomsbury, 2024.
2. S. K. Gupta et al. (eds.), *Exploring the Impact of Extended Reality (XR) Technologies on Promoting Environmental Sustainability*, Springer, 2025.
3. N. Diakopoulos, *Automating the News: How Algorithms Are Rewriting the Media*, Harvard University Press, 2019.
4. S. Madary and T. K. Metzinger, *Real Virtuality: A Code of Ethical Conduct for Virtual and Augmented Reality*, Frontiers in Robotics and AI
5. R. Calo, *Artificial Intelligence Policy: A Primer and Roadmap*, UC Davis Law Review (selected readings).

Annexure-2

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

(Course Structure and Curriculum of B.Tech. (ECE) Fourth Year)

Faculty of Technology
University of Delhi

Detailed Course Structure and Curriculum of B.Tech. (ECE) Fourth Year

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Course Structure of B. Tech (ECE) Fourth Year
Fourth Year

Semester – VII

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-19	Optical and Wireless Communication	3	0	1	4
2.	DSE-5	Select a course from the specified list of DSE-5				4
3.	DSE-6/GE-7	Select a course from the specified list of DSE-6/GE-7				4
4.	DSE-7/GE-8	Select a course from the specified list of DSE-7/GE-8				4
5.	Dissertation on Major or Dissertation on Minor / Specialisation or Academic Project / Entrepreneurship					6
Total Credits						22

Semester – VIII

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-20	RF and Microwave Engineering	3	0	1	4
2.	DSE-8	Select a course from the specified list of DSE-8				4
3.	DSE-9/GE-9	Select a course from the specified list of DSE-9/GE-9				4
4.	DSE-10/GE-10	Select a course from the specified list of DSE-10/GE-10				4
5.	Dissertation on Major or Dissertation on Minor / Specialisation or Academic Project / Entrepreneurship					6
Total Credits						22

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week
T (01 Credit) is equivalent to 01 contact hour per week
P (01 Credit) is equivalent to 02 contact hours per week

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department

S. No.	Semester	DSE	Course Title
1.	VII	DSE – 5	Advanced DSP
2.			Digital ASIC Design: RTL to GDS - II
3.		DSE – 6	Analog Circuit Design: Modeling and Analysis
4.			Flexible Electronics and Wearable Technology
5.		DSE – 7	Introduction to Nonlinear Dynamics
6.			Verilog Ecosystem
7.	VIII	DSE – 8	Microwave Devices and Circuits
8.			Scientific Computing for Electronics
9.		DSE – 9	Optoelectronic Devices and Systems
10.			Introduction to Quantum Computing
11.		DSE – 10	AI for Wireless Communication
12.			Low Power VLSI Design

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Specialization and Minors offered by the Department

Semester	DSE/ GE	ECE Minor (Open only for CSE/ EE)	Specializations for ECE/ Minors for EE and CSE			
			Telecommunication Networks	**VLSI Technology and System Design**	**IoT System Design**	**Computer Vision**
VII	DSE-6/ GE-7	Applied MIMO and B5G Physical Layer Design	Photonic and Optical Communication Networks	Microwave Integrated Circuits	Edge IoT Prototyping with RTOS	Medical Image Processing
	DSE-7/ GE-8	Sensors and Actuators for Smart Systems	Optimization for Engineering	Analog VLSI Filter Design	Industry 4.0 and IIoT	Remote Sensing and Satellite Image Processing
VIII	DSE-9/ GE-9	Statistical Signal Processing	Wireless Systems Implementation	Introduction to Nonlinear Circuits	IoT Sensor and Data Integration	Forensic Image Processing
		Speech and Video Processing	Smart Antennas, Beamforming and OTA Testing for 5G/6G	RF Integrated Circuit Design	IoT Reliability Engineering	Computer Graphics
	DSE-10/ GE-10	Software Defined Radio and Waveform Engineering	Enabling Technologies for 6G Communication	Semiconductor Device Characterization	LPWAN IoT Networks	Computational Photography and Mobile Imaging
		Compressive Sensing and Sparse Representation	Estimation for Wireless Communication	Compact Modeling with Verilog-A	Edge Computing	Image Restoration and Inpainting

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester VII

Optical and Wireless Communication (DSC - 19)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Optical and Wireless Communication	4	3	0	1	Concepts of Digital Communication Systems

Course Objectives:
- Understand the principles of light propagation in optical fibers and signal degradation mechanisms.
- Design and analyze optical communication links considering power and bandwidth requirements.
- Characterize wireless channels and apply appropriate modulation and multiple access techniques.
- Evaluate performance of cellular and modern wireless communication systems.
- Apply knowledge of current and emerging optical and wireless technologies.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand the fundamentals and architecture of optical and wireless communication systems.
2. Analyze optical fiber transmission, sources, detectors, and system impairments.
3. Explain wireless propagation mechanisms, fading, and noise effects.
4. Design and analyze basic optical and wireless communication links.
5. Evaluate the performance of modern communication systems used in real-world applications.

Unit – I **(11 hours)**
Introduction and Fundamentals of Optical Communication: Overview of communication systems and evolution of optical and wireless communications, Electromagnetic spectrum and applications, Principle of light propagation in optical fibers, Ray theory, modes of propagation, Step-index and graded-index fibers, Numerical aperture and acceptance angle, Types of optical fibers, Optical losses: absorption, scattering, bending losses, Dispersion: intermodal, material, and waveguide dispersion.

Unit – II **(12 hours)**
Optical Communication Components and Systems: Optical sources: LEDs and laser diodes, Characteristics and modulation techniques, Optical detectors: PIN and APD photodiodes, Receiver characteristics and noise sources, Optical amplifiers (EDFA, Raman amplifiers), Optical couplers, connectors, and splices, Optical transmitter and receiver design, Digital optical communication systems, Wavelength Division Multiplexing (WDM), Optical link power budget and rise-time budget, Free Space Optical (FSO) communication, Introduction to optical networks and passive optical networks (PON).

Unit – III **(11 hours)**

Fundamentals of Wireless Communication and Channel Modeling: Basics and architecture of wireless communication systems, Wireless channel characteristics, Propagation mechanisms: reflection, diffraction, scattering, Path loss models: free space, log-distance, Okumura-Hata, Noise and interference in wireless systems, Signal-to-noise ratio and channel capacity, Large-scale and small-scale fading, Statistical fading models: Rayleigh, Rician, Nakagami, Doppler effect and coherence time.

Unit – IV **(11 hours)**

Modern Wireless Systems and Emerging Technologies: Diversity techniques: time, frequency, and space diversity, Multiple-input multiple-output (MIMO) systems, Cellular communication concepts, Multiple access techniques: FDMA, TDMA, CDMA, OFDMA, Overview of 4G LTE and 5G systems, Optical wireless communication: VLC/LiFi, Hybrid optical–wireless systems, Millimeter-wave, terahertz communication, and 6G vision, Security challenges in optical and wireless communication.

Essential Readings

1. Wireless Communications by A. Goldsmith (Cambridge University Press)
2. Optical Fiber Communications by G. Keiser (McGraw-Hill)
3. Wireless Communications: Principles and Practice by T. S. Rappaport (Pearson)

Suggested Readings

1. Modern Control Systems by Richard C. Dorf and Robert H. Bishop (Pearson)
2. Optical Fiber Communications: Principles and Practice by J. M. Senior and M. Y. Jamro (Pearson)
3. Modern Wireless Communications by S. Haykin and M. Moher (Pearson)

List of Experiments (Hardware and Software using Matlab / Simulink – at least 5) **(30 hours)**

1. Simulation of numerical aperture and acceptance angle of step-index and graded-index optical fibers.
2. MATLAB-based analysis of attenuation and bending losses in optical fiber links.
3. Modeling and performance analysis of LED and Laser Diode characteristics for optical communication.
4. Simulation of optical receiver using PIN and APD photodiodes and noise analysis.
5. Optical link power budget and rise-time budget analysis using MATLAB.
6. Simulation and performance evaluation of WDM-based optical communication systems.
7. MATLAB simulation of wireless path loss models (Free-space, Log-distance, Okumura–Hata).
8. Simulation and analysis of wireless fading channels (Rayleigh, Rician, Nakagami).
9. BER and SNR performance analysis of digital modulation schemes (BPSK, QPSK) over wireless channels.
10. Performance evaluation of diversity and MIMO techniques in wireless communication using MATLAB.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester VII

Advanced DSP (DSE - 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Advanced DSP	4	3	0	1	Concepts of Signals and Systems, Digital Signal Processing

Course Objectives:
- To study the fundamentals of multi-rate signal processing.
- To study the different linear prediction and optimum linear filters.
- To study different adaptive filters and Algorithms.
- To study the various nonparametric methods for power spectrum estimation.
- To study the various parametric methods for power spectrum estimation.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand the fundamentals of sampling theory and multi-rate signal processing.
2. Understand the different linear predictions and optimum linear filters.
3. Understand the different adaptive filters.
4. Analyse the various nonparametric methods for power spectrum estimation.
5. Analyse the various parametric methods for power spectrum estimation.

Unit – I **(11 hours)**
Review of discrete-time signals and systems, DTFT and DFT; aliasing and band-limited signals. Sampling theory: bandpass sampling, reconstruction. Mult-irate systems: up-sampling and down-sampling, time-domain expressions, spectra of interpolated and decimated signals. Polyphase decomposition, efficient structures for decimators and interpolators, multistage decimation/interpolation, applications in sub-band coding and filter banks.

Unit – II **(12 hours)**
Linear Prediction and Optimum Linear Filters: Innovations Representation of a Stationary Random Process, Forward and Backward linear prediction, Solution of the Normal Equations, Properties of linear prediction-Error Filter, AR Lattice and ARMA Lattice-Ladder Filters.
Adaptive Filters: Adaptive direct-form FIR Filters- LMS Algorithms, Adaptive Direct-Form Filter- RLS Algorithms, Adaptive Lattice-Ladder Filters.

Unit – III **(11 hours)**
Power Spectral Estimation: Estimation of Spectra from Finite Duration Observations of a Signal, Periodogram, Nonparametric Methods for Power Spectral Estimation: Bartlett, Welch, Blackman and Tukey methods, Comparison of performance of Non-Parametric Power Spectrum Estimation Methods.

Unit – IV **(11 hours)**
Parametric Methods for Power spectrum estimation: Relationship between Auto-correlation and Model Parameters, Yule-Walker method for AR model parameters, Burg method for AR model parameter, Unconstrained Least Squares Methods for AR model parameter, Sequential Estimation methods for AR model parameter, Moving Average (MA) and ARMA Models Minimum Variance Method.

Essential Readings

1. Digital Signal Processing: Principles, Algorithms and Applications by J. G. Proakis and D. G. Manolakis (Pearson)
2. Statistical Digital Signal Processing and Modeling by Monson H. Hayes (Wiley)
3. Adaptive Filter Theory by Simon Haykin (Pearson)

Suggested Readings

1. Discrete-Time Signal Processing by A. V. Oppenheim and R. W. Schafer (Pearson)
2. Digital Signal Processing: A Computer-Based Approach by S. K. Mitra (McGraw Hill)
3. Multirate Systems and Filter Banks by P. P. Vaidyanathan (Prentice Hall)
4. Algorithm Collections for Digital Signal Processing Applications Using Matlab by E. S. Gopi (Springer)

List of Experiments (Hardware and Software using Matlab/Simulink) **(30 hours)**

1. Generation and visualization of basic discrete time signals; verification of sampling/aliasing using different sampling rates.
2. Implementation of up sampler, down sampler, and combined multistage rate converters; spectrum plots before/after rate change.
3. Design and implementation of FIR interpolation and decimation filters using window methods; verification of polyphase structure.
4. Estimation of PSD using periodogram, Bartlett, and Welch methods; comparison of resolution and variance.
5. AR model parameter estimation (Yule–Walker/Burg) and parametric PSD.
6. Design and simulation of an FIR Wiener filter for noise reduction in a noisy signal.
7. LMS adaptive filter for system identification or echo cancellation; study of step size effect on convergence.
8. RLS adaptive filter for channel equalization; comparison with LMS in terms of convergence speed.
9. Implementation of a chosen FIR/IIR or adaptive filter in real time on a DSP or embedded board

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Digital ASIC Design: RTL to GDS - II (DSE – 5)
(Credit Distribution and Prerequisites of the Course)

<table>
<tr><th rowspan="2">Course Title</th><th rowspan="2">Credits</th><th colspan="3">Credit Distribution of the Course</th><th rowspan="2">Prerequisite of the course (if any)</th></tr>
<tr><th>Lecture</th><th>Tutorial</th><th>Practical</th></tr>
<tr><td>Digital ASIC Design: RTL to GDS - II</td><td>4</td><td>2</td><td>0</td><td>2</td><td>Concepts of Digital Electronics / Digital System Design</td></tr>
</table>

Course Objectives:

- To develop practical RTL design skills using Verilog / VHDL for synthesizable digital hardware blocks.
- To introduce hands - on ASIC flow from RTL → netlist → floorplan → place → route → GDS-II using an EDA toolchain.
- To build ability to run and automate the flow using Tcl scripting (batch runs, parameter sweeps, reproducible builds).
- To develop practical verification skills using testbenches, and waveform debugging techniques.
- To expose students to FPGA prototyping as a hardware validation step.

Course Outcomes:

After completing the course, the students should be able to:

1. Write synthesizable Verilog / VHDL RTL for combinational / sequential systems, including FSMs and datapath / control modules.
2. Create self-checking testbenches, debug using waveforms, and validate corner cases with a structured verification workflow.
3. Run the RTL-to-GDS-II flow using industry-style steps (synthesis, floorplan, placement, CTS (overview), routing) and generate GDS-II outputs.
4. Interpret key reports: area, timing slack, power estimate, utilization, congestion and apply practical fixes (pipelining, constraints, floorplan tweaks).
5. Prototype RTL on FPGA and correlate hardware behavior with simulation results.

Unit – I **(15 hours)**

RTL Design Fundamentals: RTL constructs, combinational vs sequential. Adder, Counters. FSM, debouncing, and clock design.

Synthesis: RTL to gates / standard cells, optimization goals (timing / area). Constraints: clocks, input / output delays, false paths / multicycle. Timing essentials: critical path, setup / hold, slack. Gate-level simulation.

Unit – II **(15 hours)**

Physical Design Concepts: floorplanning, placement, clocking concept, routing, and generating final layout GDS-II. Congestion, wirelength trends, timing after placement / route, and area utilization.

Tcl scripting: running steps, setting variables. Parameter sweeps: clock period vs area / timing. STA / DRC / LVS checks. FPGA: Map RTL block to FPGA I/O (LED) for Hardware Validation.

Essential Readings

1. Digital Design and Computer Architecture by David Harris & Sarah Harris (Morgan Kaufmann)
2. FPGA Prototyping by Verilog Examples by Pong P. Chu (Wiley)
3. RTL Hardware Design Using VHDL by Pong P. Chu (Wiley)

Suggested Readings

1. Yosys Documentation (Official)
2. OpenROAD / OpenLane Documentation (Official)
3. KLayout / Magic Documentation (Official)

List of Experiments (Hardware based on FPGA boards / Software based on Open Source tools / Cadence)
(60 hours)

1. Write synthesizable RTL in Verilog/VHDL and verify reset/enable using simulation.
2. Design an FSM in Verilog / VHDL and validate state transitions with a testbench.
3. Implement a small datapath block and verify outputs using waveforms/pass-fail checks.
4. Run synthesis to generate a gate-level netlist and basic area/timing reports.
5. Improve timing by RTL optimization and compare before/after synthesis results.
6. Perform floorplanning by varying utilization/aspect ratio.
7. Run place-and-route to produce the final layout and GDS-II.
8. To use a TCL script for automating key steps.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Analog Circuit Design: Modeling and Analysis (DSE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Analog Circuit Design: Modeling and Analysis	4	3	0	1	Concepts of Electronic Devices and Circuits

Course Objectives:

- To understand the operation of four terminal MOS device and its use in multistage amplifier.
- To inculcate the ability to analyse frequency response and determine poles and zeros.
- To learn the essentials of current mirror and operational amplifier and their analysis
- To understand the various types of noise in amplifiers and their interference.

Course Outcomes:

After completing the course, the students should be able to:

1. Analyze MOS models and multistage amplifiers using Miller's theorem/Bode plots, while understanding the issues related to stability/noise.
2. Perform high/low-frequency analysis of MOS amplifiers using time-constant methods for bandwidth estimation.
3. Design CMOS op-amps and current mirrors with stability compensation.
4. Understand noise related issues, perform its modelling, compute S/N ratios, and evaluate harmonic distortion in amplifiers.

Unit – I **(12 hours)**

Basic MOS models – Low Frequency and High Frequency, Bulk Effect, Second order effects, short channel effects, multistage amplifiers and their small-signal analysis, cascade and cascode amplifier, folded cascode amplifier, Millers theorem, Bode plot, stability and noise issues in amplifiers.

Unit – II **(15 hours)**

High Frequency analysis of Single stage amplifiers (CS, CG, CD, CS amplifiers with source degeneration) with conventional and time constant methods. Dominant pole approach, Low frequency analysis of CS amplifier with time constant method. High frequency analysis of cascode amplifier. Step response of amplifier. Short circuit transconductance of common source amplifier, Input and output impedance of amplifier, bandwidth estimation.

Unit – III **(10 hours)**

Basic current mirrors, Wilson and Cascode current mirrors, CMOS Operational amplifiers: Single-Stage Op-Amps, Telescopic cascode Op-Amp, Folded-Cascode Op-Amps, Two-stage Op-Amps, Stability and Compensation.

Unit – IV **(08 hours)**

Signals, Information, Interference and noise, Statistical Characteristics of Noise, Types of Noise, S/N ratio, noise model for circuit analysis, input referred noise, noise analysis of amplifiers, total harmonic distortion.

Essential Readings

1. Analog MOS integrated circuits for signal processing by Gregorian, Roubik, and Gabor C. Temes (Wiley).
2. Microelectronic Circuits by Sedra, Adel S., and Kenneth C. Smith (Oxford University Press).
3. Design of Analog CMOS Integrated Circuits by Behzad Razavi (Tata McGraw-Hill)

Suggested Readings

1. CMOS Analog Circuit Design by Allen, Phillip E., and Douglas R. Holberg (Oxford University Press).
2. Microelectronics: Digital and Analog Circuits and Systems by Millman, Jacob, and Arvin Grabel (McGraw-Hill).
3. Analog integrated circuit design by Johns, David A., and Ken Martin (John Wiley & Sons).
4. Analysis and design of analog integrated circuits by Gray, Paul R., et al (Wiley).
5. CMOS Digital Integrated Circuits Analysis & Design by Sung-Mo Kang, Yusuf Leblebici, (McGraw-Hill)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Plot DC transfer curves for NMOS / PMOS (Id vs Vgs, Vds) showing cutoff, triode, saturation regions with short-channel effects (use 0.18μm models).
2. Compare voltage gain (Av), output impedance, and frequency response of basic NMOS gain stage vs cascode amplifier.
3. Simulate gain, phase margin, unity-gain bandwidth, and dominant pole location for Folded Cascode Amplifier.
4. Compare output impedance and compliance range of basic, Wilson, and cascode current mirrors.
5. Design and simulate open-loop gain, GBW, phase margin, and slew rate. Include stability analysis with unit-step response.
6. Implement Two-Stage Op-Amp with Miller compensation capacitor; plot phase margin improvement and settling time. Compare compensated vs uncompensated.
7. Calculate total RMS noise, input-referred noise voltage, and SNR for a cascode amplifier.
8. Simulate common-mode vs differential gain with active load. Plot CMRR vs frequency using dual-tone inputs.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Flexible Electronics and Wearable Technology (DSE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Flexible Electronics and Wearable Technology	4	3	0	1	Concepts of Electrical and Electronics Engineering / Electronic Devices and Circuits

Course Objectives:

- To introduce materials, device structures, and processes used in flexible and wearable electronics.
- To design basic flexible components (interconnects, sensors, TFT-style structures).
- To fabricate low cost flexible devices and characterize electrical performance using SMU / LCR / DMM setups.
- To model flexible structures and device behavior using EDA tools and correlate with measurements.

Course Outcomes:

After completing the course, the students should be able to:

1. Select suitable flexible substrates, conductors, and encapsulation for a wearable use case.
2. Fabricate basic flexible interconnects / sensors using low-cost methods and measure key electrical metrics and repeatability.
3. Perform bending / fatigue tests and quantify performance degradation.
4. Use EDA tools to simulate strain / electrical behavior and explain correlation gaps between model and measurements.

Unit – I **(10 hours)**

Basics of Flexible & Wearable Electronics: Applications and form factors (patch, band, textile), substrates (PET, PI/Kapton, TPU), conductors, encapsulation, wearable constraints and simple design rules.

Unit – II **(11 hours)**

Fabrication & Interconnect Design: Patterning methods (copper-tape, stencil/screen printing, vinyl-mask), curing, flexible interconnect (straight vs serpentine), connectors and contact resistance, packaging.

Unit – III **(12 hours)**

Wearable Sensors: Strain sensors (piezoresistive), pressure sensors (piezoresistive / capacitive), temperature sensing (RTD / NTC), basic bioelectrodes (ECG / EMG overview), signalling /calibration.

Unit – IV **(12 hours)**

Characterization: SMU / LCR / DMM (R–t, I–V, C–V), two-wire vs Kelvin sensing, bending / fatigue test protocols, failure modes, EDA based bending/strain hotspot simulation, EDA based device simulation.

Essential Readings

1. Flexible Electronics: Materials and Applications by William S. Wong & Alberto Salleo (Springer)
2. Wearable Sensors: Fundamentals, Implementation and Applications by Edward Sazonov & Michael R. Neuman (Academic Press)
3. Printed Electronics: Materials, Technologies and Applications by Zheng Cui (Wiley)

Suggested Readings

1. Stretchable Electronics by Takao Someya (Wiley)
2. Bioelectronics: From Theory to Applications edited by Itamar Willner and Eugenii Katz (Wiley, 2005)
3. Flexible Electronics: From Materials to Devices by Guozhen Shen and Zhiyong Fan (World Scientific)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Build and test flexible interconnects and measure basic electrical resistance and contacts.
2. Compare different trace geometries under bending and repeated flexing.
3. Develop a basic strain sensor and evaluate sensitivity and repeatability.
4. Model and optimize strain sensing structures using simulation / EDA tools.
5. Model and optimize pressure-sensing structures using simulation / EDA tools.
6. Prototype a pressure sensor and characterize key performance metrics.
7. Prototype a temperature sensor and perform calibration and stability checks.
8. Develop wearable electrode structures and evaluate electrical interface quality.
9. Simulate multilayer flexible stacks to identify high-stress regions and failure risks.
10. Perform simple reliability testing and track performance degradation trends.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Introduction to Nonlinear Dynamics (DSE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Nonlinear Dynamics	4	3	0	1	NIL

Course Objectives:

- Introduce the fundamental concepts of nonlinear dynamical systems and state-space analysis.
- Develop qualitative and quantitative understanding of one- and two-dimensional nonlinear systems.
- To train students in phase-line and phase-plane methods for analysing system behaviour.
- Provide a rigorous foundation in linearization and local stability theory.
- Explain how bifurcations arise in parameter-dependent systems.
- Introduce the mathematical mechanisms leading to chaos.
- Familiarize students with nonlinear oscillators and driven systems.
- To build analytical and computational skills for studying real physical and engineering systems.

Course Outcomes:

After completing the course, the students should be able to:

1. Formulate dynamical systems and analyze one-dimensional nonlinear flows.
2. Apply linearization methods to determine local stability of equilibria.
3. Construct and interpret phase-plane portraits of two-dimensional systems.
4. Classify bifurcations and identify routes to chaos in nonlinear systems.
5. Analyze nonlinear oscillators and chaotic systems using analytical and numerical tools.

Unit – I **(10 hours)**

Dynamical systems and state space; autonomous first-order differential equations; fixed points and phase-line analysis; stability of equilibria; nonlinear flows; physical interpretation and potential analogy; global versus local behavior; examples from population dynamics, reaction kinetics, and particle motion.

Unit – II **(10 hours)**

Linear systems in one dimension; exponential growth and decay; relaxation time scales; linearization of nonlinear systems; Jacobian in one dimension; local stability criteria; marginal stability and breakdown of linear models; influence of nonlinear terms; critical slowing down.

Unit – III **(10 hours)**

Planar dynamical systems; phase space and trajectories; linear systems in two dimensions; classification of fixed points (nodes, saddles, spirals, centers); eigenvalues and eigenvectors; nullclines and graphical methods; conservative and dissipative systems; limit cycles; Poincaré–Bendixson theorem; nonlinear oscillators (Van der Pol, Duffing type systems).

Unit – IV **(15 hours)**

Parameter-dependent systems; local bifurcations: saddle-node, transcritical, pitchfork and Hopf bifurcations; normal forms and universality; routes to chaos; logistic map and iterated maps; period-doubling cascade; bifurcation diagrams; sensitive dependence on initial conditions; strange attractors; Poincaré sections; chaos in driven nonlinear oscillators.

Suggested Readings

1. Nonlinear Dynamics and Chaos by Steven H. Strogatz (CRC Press)
2. Deterministic Chaos by H. G. Schuster and W. Just (Wiley-VCH)
3. Chaos in Dynamical Systems by Edward Ott (Cambridge University Press)

Essential Readings

1. Chaos: An Introduction to Dynamical Systems by Kathleen Alligood, Tim Sauer, and James Yorke (Springer)
2. Nonlinear Dynamics and Chaos by J. M. T. Thompson and H. B. Stewart (Wiley)

List of Experiments (Software using Matlab) **(30 hours)**

1. Phase-line simulations.
2. Numerical integration of nonlinear ODEs.
3. Phase-plane plotting; limit-cycle detection.
4. Construction of bifurcation diagrams.
5. Poincaré maps; Lyapunov exponent estimation.
6. Chaos in Duffing and driven oscillator models.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Verilog Ecosystem (DSE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Verilog Ecosystem	4	2	0	2	Basic Programming Skills and Fundamentals of Digital Electronics / Digital System Design

Course Objectives:

- To develop practical coding skills in Verilog HDL including synthesizable RTL and simulation-only constructs.
- To introduce SystemVerilog for modern RTL, interfaces, packages, and verification features.
- To expose students to SystemVerilog Assertions (SVA) and basic coverage for bug-catching.
- To provide working exposure to Verilog-A / Verilog-AMS for behavioral modeling of analog / mixed-signal blocks.

Course Outcomes:

After completing the course, the students should be able to:

1. Write clean synthesizable RTL using Verilog HDL and distinguish synthesizable vs testbench-only constructs.
2. Use SystemVerilog features (always_ff/always_comb, enum, struct, interface, package) to write safer, scalable RTL.
3. Develop verification testbenches using SystemVerilog (classes-light, constrained random basics, functional coverage basics) and debug using waveforms/logs.
4. Write simple SystemVerilog Assertions (SVA) to check protocols/invariants and improve debug productivity.
5. Create basic Verilog-A/AMS behavioral models (e.g., diode/opamp/RC/sensor) and run mixed-signal style simulations at a conceptual level.

Unit – I **(15 hours)**

Verilog RTL: modules/ports/parameters, nets vs reg/logic, timescale; synthesizable combinational vs sequential coding, basic datapath blocks, and simulation basics ($display/$monitor, waveform dump).

Modern Verilog: generate blocks, parameters/localparams, functions/tasks; synthesizable coding, basic clock/reset strategy, and gate-level netlist concept.

Unit – II **(15 hours)**

SystemVerilog: always_comb/ff, typedef/enum/struct, packed arrays, interfaces/packages; basic verification and assertions (simple).

Verilog-A fundamentals: analog behavioral modeling approach, simple continuous-time models (RC / diode / opamp), and a high-level view of mixed-signal blocks (ADC/DAC).

Essential Readings

1. SystemVerilog for Design: A Guide to Using SystemVerilog for Hardware Design and Modeling by Stuart Sutherland, Simon Davidmann, and Peter Flake (Springer)
2. The Verilog® Hardware Description Language by Donald E. Thomas and Philip R. Moorby (Springer)
3. Verilog-A: Analog Behavioral Modeling by Ken Kundert and Olaf Zinke (Springer).

Suggested Readings

1. IEEE 1364 Verilog HDL Standard and IEEE 1800 SystemVerilog Standard.
2. FPGA Prototyping by SystemVerilog Examples: Xilinx MicroBlaze MCS SoC Edition by Pong P. Chu (Wiley)
3. SystemVerilog for Verification: A Guide to Learning the Testbench Language Features by Chris Spear and Greg Tumbush (Springer)

List of Experiments (Hardware / Software) **(60 hours)**

1. Write an RTL module and verify using simulation and waveforms.
2. Implement sequential RTL and validate timing behavior in simulation.
3. Design an FSM and verify state transitions using a testbench.
4. Use parameterization/generate constructs to build scalable N-bit logic.
5. Write RTL using SystemVerilog constructs.
6. Create and use a simple interface to connect DUT and testbench.
7. Develop a SystemVerilog testbench and run directed tests for a small DUT.
8. Build a reusable verification setup (non-UVM) and adapt it across multiple DUTs.
9. Create simple Verilog-A behavioral models and validate expected responses.
10. Set up a basic mixed-signal simulation with digital stimulus interacting with an analog model.

(Note: Course instructor may add/ delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester VIII

RF and Microwave Engineering (DSC - 20)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
RF and Microwave Engineering	4	3	0	1	Concepts of Electromagnetic Theory

Course Objectives:

- To make students comfortable with RF by focusing on measurement and tuning.
- To teach core industry skills: S-parameters, VNA use, matching networks, RF prototyping.
- To design, fabricate, and test real antennas and RF blocks relevant to IoT/Wi-Fi/ISM bands.
- To introduce practical EMI/EMC debugging and RF system thinking with link budgets and RSSI validation.

Course Outcomes:

After completing the course, the students should be able to:

1. Measure and interpret S11/S21, VSWR, return loss, and apply them to improve RF designs.
2. Design and validate matching networks and simple RF filters using simulation + VNA measurement.
3. Design, prototype, and tune a 2.4 GHz antenna and evaluate performance changes due to enclosure / hand effects.
4. Perform an RF link budget and verify predictions using RSSI/measurement logs.

Unit – I **(11 hours)**

RF Measurements: RF frequency view (phasors, complex impedance/admittance, dB/dBm, noise floor intuition), two-port network for RF circuits, S – parameters, return loss / VSWR / bandwidth, VNA measurement, error sources, repeatability, time-domain view using VNA.

Unit – II **(12 hours)**

Transmission Lines & Matching: Distributed vs lumped behavior, transmission line parameters, impedance transformation, Smith chart basics, networks: L-network / π/T / stub matching / quarter-wave transformer, matching for antennas vs matching for amplifiers, stability / oscillation, and RF layout matching.

Unit – III **(12 hours)**

Planar RF Blocks: Planar transmission, microstrip design for 50Ω, discontinuities (bends / steps / tees), resonators (λ/4, λ/2), filter design (fc, BW, ripple, IL), couplers and power dividers (Wilkinson), RF switches / attenuators, mixers and frequency, oscillators / PLLs and phase noise.

Unit – IV **(10 hours)**
Antennas, System Integration and EMI/EMC: Monopole / dipole / patch tuning by S11, antenna efficiency, Friis, RSSI, EMI / EMC coupling paths, shielding / grounding.

Essential Readings

1. RF Circuit Design: Theory and Applications by Reinhold Ludwig and Gene Bogdanov (Pearson)
2. Microwave Transistor Amplifiers: Analysis and Design by Guillermo Gonzalez (Pearson)
3. Microwave Engineering by David M. Pozar (Wiley)

Suggested Readings

1. Antenna Theory: Analysis and Design by Constantine A. Balanis (Wiley)
2. Foundations for Microwave Engineering by Robert E. Collin (McGraw-Hill)
3. ARRL Handbook (selected practical RF measurement sections)

List of Experiments (Hardware / Software using Matlab/Simulink – at least 5) **(30 hours)**

1. Use a VNA to measure basic RF components (loads/cables/antennas) and perform calibration.
2. Interpret S-parameters (return loss/VSWR) for impedance matching quality.
3. Design and validate a simple matching network to improve S11.
4. Implement stub-based matching and verify tuning impact through measurement.
5. Design and test a controlled-impedance transmission line and measure insertion loss (S21).
6. Build and characterize a basic RF filter and measure bandwidth/attenuation.
7. Design, tune, and evaluate a 2.4 GHz antenna to meet a target match level.
8. Perform basic RF system checks using range tests, link budget comparison, and spectrum/EMI measurements.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester VIII

Microwave Devices and Circuits (DSE – 8)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Microwave Devices and Circuits	4	3	0	1	Concepts of Electromagnetic Theory and Communication Systems

Course Objectives:

- Introduce fundamental concepts of microwave passive and active devices.
- Understand the operation of microwave semiconductor devices and tubes.
- Develop knowledge of microwave circuit design (matching, filters, amplifiers).
- Familiarize students with microwave measurement techniques and instruments.
- Expose students to microwave systems and modern applications.

Course Outcomes:
After completing the course, the students should be able to:

1. Explain the principles and characteristics of microwave passive and active components.
2. Analyze microwave semiconductor devices and vacuum tubes.
3. Apply microwave design techniques for matching, filters, amplifiers, oscillators, and antennas.
4. Perform and interpret microwave measurements using standard instruments.
5. Describe microwave systems such as radar, satellite communication, navigation, and RFID.
6. Understand modern microwave technologies including MMICs, RF MEMS, imaging, and EMI/EMC.

Unit – I **(11 hours)**
Passive and Active Microwave Devices: Microwave passive components: Directional Coupler, Power Divider, Magic Tee, Attenuator, Resonator. Microwave active components: Diodes, Transistors, Oscillators, Mixers. Microwave Semiconductor Devices: Gunn Diodes, IMPATT diodes, Schottky Barrier diodes, PIN diodes. Microwave Tubes: Klystron, TWT, Magnetron.

Unit – II **(12 hours)**
Microwave Design Principles: Impedance transformation, Impedance Matching, Microwave Filter Design, RF and Microwave Amplifier Design, Microwave Power Amplifier Design, Low Noise Amplifier Design, Microwave Mixer Design, Microwave Oscillator Design. Microwave Antennas- Antenna parameters, Antenna for ground-based systems, Antennas for airborne and satellite borne systems, Planar Antennas.

Unit – III **(11 hours)**

Microwave Measurements: Power, Frequency and impedance measurement at microwave frequency, Network Analyzer and measurement of scattering parameters, Spectrum Analyzer and measurement of spectrum of a microwave signal, Noise at microwave frequency and measurement of noise figure. Measurement of Microwave antenna parameters.

Unit – IV **(11 hours)**

Microwave Systems: Radar, Terrestrial and Satellite Communication, Radio Aidsto Navigation, RFID, GPS. Modern Trends in Microwaves Engineering- Effect of Microwaves on human body, Medical and Civil applications of microwaves, Electromagnetic interference and Electromagnetic Compatibility (EMI & EMC), Monolithic Microwave ICs, RFMEMS for microwave components, Microwave Imaging.

Essential Readings

1. Microwave Engineering by David M. Pozar (John Wiley & Sons, 4th Edition)
2. Microwave Devices and Circuits by Samuel Y. Liao (Pearson / PHI, 3rd Edition)
3. RF Circuit Design: Theory and Applications by Christopher Bowick (Elsevier)

Suggested Readings

1. Foundations for Microwave Engineering by Robert E. Collin, (Wiley India, 2nd Edition)
2. Microwave Engineering by Annapurna Das and Sisir K. Das (Tata McGraw-Hill)
3. Microwave Circuits and Passive Devices by M. L. Sisodia and G. S. Raghuvanshi (New Age International)
4. Microwave and Radar Engineering by M. Kulkarni (Umesh Publications)

<u>List of Experiments</u> (Hardware / Software – at least 5) **(30 hours)**

1. Study of various microwave components and instruments.
2. Measurement of guide wavelength and frequency using a X-band slotted line setup.
   Measurement of low and high VSWR using a X-band slotted line setup.
3. Draw the V-I characteristics of a Gunn diode and determine the output power and frequency as a function of voltage.
4. Study the square wave modulation of microwave signal using PIN diode.
5. Study and measure the power division and isolation characteristics of a microstrip 3dB power divider.
6. Study of rat race hybrid ring (equivalent of waveguide Magic-Tee) in micro-strip.
7. To study the characteristics of micro-strip 3dB branch line coupler, strip line backward wave coupler as a function of frequency and compare their bandwidth.
8. Measure the microwave input, direct, coupled, and isolated powers of a backward wave strip line coupler at the centre frequency using a power meter. From the measurements calculate the coupling, isolation, and directivity of the coupler.
9. Study of field pattern of various modes inside a rectangular and circular waveguide.
10. Find the change in characteristics impedance and reflection coefficients of the transmission line by changing the dielectric properties of materials embedded between two conductors.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Scientific Computing for Electronics (DSE – 8)
(Credit Distribution and Prerequisites of the Course)

<table>
<tr><th rowspan="2">Course Title</th><th rowspan="2">Credits</th><th colspan="3">Credit Distribution of the Course</th><th rowspan="2">Prerequisite of the course (if any)</th></tr>
<tr><th>Lecture</th><th>Tutorial</th><th>Practical</th></tr>
<tr><td>Scientific Computing for Electronics</td><td>4</td><td>3</td><td>0</td><td>1</td><td>NIL</td></tr>
</table>

Course Objectives:

- To introduce students to Linear Programming techniques for optimal allocation of resources.
- To develop understanding of transportation and network models used in operations research.
- To apply project management techniques such as CPM and PERT for planning and scheduling.
- To familiarize students with decision-making methods under uncertainty, risk, and conflict.
- To build analytical skills for solving real-life managerial and optimization problems.

Course Outcomes:

After completing the course, the students should be able to:

1. Formulate and solve Linear Programming Problems using graphical and simplex methods.
2. Solve transportation and network optimization problems using standard techniques.
3. Analyze project scheduling using CPM and PERT, including floats and time-cost trade-off.
4. Apply decision theory models to make decisions under uncertainty and risk.
5. Solve game theory problems using maximin, minimax, dominance rule, and graphical methods.

Unit – I **(11 hours)**

Linear Programming: Formulation of L.P. Problems, Graphical Solutions (Special cases: Multiple optimal solution, infeasibility, unbounded solution); Simplex Method, Special cases, Big-M method and Two-phase method; Duality (emphasis on formulation & economic interpretation);

Unit – II **(11 hours)**

Transportation: Formulation of Transportation Problems, Solution by N.W. Corner Rule, Least Cost method, Vogel's Approximation Method (VAM), Modified Distribution Method; Special cases: Multiple Solutions, Maximization case, unbalanced case, prohibited routes.

Unit – III **(10 hours)**

Network Analysis: Basic Concept, Construction of the Network diagram, Critical Path Analysis, float and slack analysis (Total float, free float, independent float), probability consideration in PERT, Time-Cost optimization in Project. (Interface with Project Management software)

Unit – IV **(13 hours)**

Decision Theory: Decision making environment, Construction of Pay off Table, Opportunity Loss Table, Decision under uncertainty, Decision under Risk: EMV, EOL, EVPI.
Decision under Conflict: Game Theory, Two person Zero-Sum games, Maximin Minimax Principle, Games without Saddle point- Mixed strategy, Dominance Rule; Reduction of m x n game and solution of 2×2, 2 x s, and r x 2 cases by Graphical Method.

Essential Readings

1. Operations Research: An Introduction by Hamdy A. Taha (Pearson)
2. Introduction to Operations Research by Hillier & Lieberman (McGraw Hill)
3. Linear Programming and Network Flows by M. S. Bazaraa and J.J. Jarvis (John Wiley & Sons)

Suggested Readings

1. Quantitative Techniques in Management by N. D. Vohra (McGraw)
2. Operations Research: Theory and Applications by J.K. Sharma (Macmillan India Limited)
3. Operations Research by Kanti Swarup, P.K. Gupta, Man Mohan (S Chand & Sons)

<u>**List of Experiments**</u> (Software using MATLAB / Python – at least 5) **(30 hours)**

1. Formulate real-life problem (production, diet, blending) into LP models and solve using Simplex method.
2. To implement two phase method for a given problem.
3. To implement Big M method for a diet problem.
4. To implement a given problem using Dual formulation and verify primal-dual relationships.
5. To analyze and implement Vogel's Approximation Method for a given transportation problem.
6. To implement optimization technique for Transportation using MODI method.
7. To construct and analyze project networks.
8. To perform scheduling analysis of project activities.
9. To analyze project completion time and uncertainty in project networks.
10. To study decision-making models and game theory concepts.
11. To design a mini project based on the latest trends in industry.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Optoelectronic Devices and Systems (DSE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Optoelectronic Devices and Systems	4	3	0	1	Introduction to Electrical and Electronics Engineering / Basics of Semiconductor Devices

Course Objectives:

- To introduce the fundamental principles of light–semiconductor interaction relevant to optoelectronic devices.
- To develop an understanding of the structure, operation, and characteristics of semiconductor optical sources, including LEDs and laser diodes.
- To explain the working principles and performance parameters of semiconductor photodetectors.
- To emphasize device physics and operating mechanisms governing optoelectronic components.
- To provide an overview of current research trends and industrial developments in optoelectronics and nanophotonics.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain optical absorption, emission, and amplification mechanisms in semiconductors.
2. Describe the structure and operation of semiconductor light sources and optical amplifiers.
3. Analyze photodetector performance and noise.
4. Understand how optoelectronic devices are integrated with electronic circuits in modern VLSI and CMOS platforms.
5. Understand emerging research directions in nanophotonics and integrated optoelectronics, and their relevance to future electronic systems.

Unit – I **(11 hours)**

Interaction of light with semiconductors. Optical absorption and emission processes. Recombination Mechanisms. Emission and absorption rates. Population inversion and optical gain in semiconductors. Principles of optical signal amplification. Material parameters affecting absorption, emission, and gain.

Unit – II **(12 hours)**

Semiconductor Light Sources: Light Emitting Diodes (LEDs): basic structure, theory of operation, spectral characteristics, efficiency, and modulation bandwidth. Semiconductor lasers: operating principle, threshold condition, resonant cavities, and device characteristics. Types of semiconductor lasers: homojunction and heterojunction lasers, quantum well lasers, DFB and DBR lasers, and VCSELs. Modulation of semiconductor lasers: direct and external modulation (conceptual). Semiconductor Optical Amplifiers (SOAs): structure, gain characteristics, and applications.

Unit – III **(11 hours)**

Semiconductor Photodetectors: Principles. Photon absorption and carrier generation. Photodetectors: PN, PIN, and avalanche photodiodes (APDs). Device structure, theory of operation, and characteristics. Responsivity, quantum efficiency, response time, and bandwidth limitations. Loss and noise mechanisms: shot noise and thermal noise (qualitative). Solar cells: principle and characteristics. Charge Coupled Devices (CCDs): structure and operation. Introduction to integrated optoelectronic detector circuits.

Unit – IV **(11 hours)**

Emerging Optoelectronic Technologies and Integration: Introduction to nanophotonic devices: quantum dots, nanowires, and photonic crystals (conceptual). Role of nanoscale confinement in enhancing emission and detection. Integrated optoelectronic circuits and CMOS-compatible photonic devices. Overview of silicon photonics, challenges in integration, scaling, power dissipation, and packaging. Future directions in optoelectronic and electronic co-integration.

Essential Readings

1. Semiconductor Optoelectronic Devices by Pallab Bhattacharya (Pearson)
2. Fundamentals of Photonics by B. E. A. Saleh and M. C. Teich (Wiley)
3. Physics of Semiconductor Devices by S. M. Sze and K. K. Ng (Wiley)

Suggested Readings

1. COMSOL Multiphysics Documentation: Wave Optics & Semiconductor Modules (COMSOL)
2. Optoelectronics and Photonics: Principles and Practices by S. O. Kasap (Pearson)
3. Silicon Photonics: An Introduction by Graham T. Reed and Andrew P. Knights (Wiley)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Measurement and analysis of the current–voltage (I–V) characteristics of a light-emitting diode (LED).
2. Experimental investigation of the relationship between applied voltage and optical output intensity of a semiconductor laser diode.
3. Characterization of diffraction grating parameters using a semiconductor laser source.
4. Experimental verification of Malus' law for plane-polarized light.
5. Experimental study and characterization of the performance characteristics of a photodetector.
6. Characterization of solar cell performance through I–V analysis and determination of fill factor and conversion efficiency.
7. Experimental study of the electro-optic effect using a semiconductor laser diode.
8. Determination of the numerical aperture of an optical fiber using a semiconductor laser source.
9. Modelling and analysis of silicon optical waveguides for evaluation of guided mode profiles and effective refractive index.
10. Simulation of nanophotonic structures such as photonic crystals or optical resonators to study enhanced light confinement.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Introduction to Quantum Computing (DSE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Quantum Computing	4	3	0	1	NIL

Course Objectives:

- Introduce the fundamental principles of quantum computing and the differences with classical computing
- Develop understanding of qubits, quantum states, superposition and entanglement
- Provide hands-on experience in designing, simulating, and executing quantum circuits
- Explain foundational quantum algorithms and their implementation
- Expose students to modern quantum computing frameworks and real quantum hardware

Course Outcomes:

After completing the course, the students should be able to:

1. Articulate the basic concepts of quantum mechanics relevant to quantum computing.
2. Represent and manipulate quantum states using Dirac notation and matrices.
3. Construct quantum circuits using standard quantum gates.
4. Implement and simulate quantum programs in Python using Qiskit.
5. Explain and demonstrate key quantum algorithms such as Deutsch, Grover, and Shor.
6. Compare quantum and classical computation through examples and real hardware execution.

Unit – I **(10 hours)**

What is quantum computing; classical vs quantum computing principles, Qubit, quantum state, binary vs quantum information, Complex vector spaces, state amplitudes and normalization, Superposition and interference, Entanglement and multi-qubit systems, Measurement and probability interpretation

Unit – II **(10 hours)**

Single-qubit gates: X, Y, Z, H (Hadamard), phase gates, multi-qubit gates: CNOT, Toffoli, SWAP, Quantum circuit notation and design, Universality of gate sets, Quantum teleportation and superdense coding, Simulation tools and circuit visualization

Unit – III **(10 hours)**

Simple algorithm examples: Deutsch and Bernstein–Vazirani, Grover's Search Algorithm (amplitude amplification), Quantum Fourier Transform (QFT) and Phase Estimation, Shor's factoring algorithm (conceptual overview and circuits), Algorithm performance and comparisons with classical counterparts

Unit – IV **(15 hours)**

Introduction to Python for quantum computing, IBM Qiskit framework: installation, basics, and workflow, Writing and simulating quantum programs, Running quantum circuits on real hardware (cloud-based IBMQ), Practical use cases and future trends in quantum computing, Optimization and error considerations in quantum circuits

Essential Readings

1. Qiskit Textbook by the Qiskit Community (Online, Open-Access)
2. Programming Quantum Computers by Eric R. Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia (O'Reilly Media)
3. Learn Quantum Computing with Python and Qiskit by Various Authors (Various Online Resources)

Suggested Readings

1. Quantum Computation and Quantum Information by Michael A. Nielsen and Isaac L. Chuang (Cambridge University Press)
2. Introduction to Quantum Mechanics by David J. Griffiths (Pearson)
3. Quantum Computing: A Gentle Introduction by Eleanor Rieffel and Wolfgang Polak (MIT Press)

List of Experiments (Hardware / Software) **(30 hours)**

1. Quantum circuits simulation using Qiskit
2. Bloch sphere visualization exercises
3. Implementation of standard quantum gates and entanglement generation
4. Grover's and Deutsch algorithms on simulators
5. Execution of simple circuits on real IBM quantum hardware
6. Projects on quantum algorithms or application case studies

(Note: Course instructor may add / delete/ update new experiments in addition to the above suggested practical exercises.)

AI for Wireless Communication (DSE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
AI for Wireless Communication	4	3	0	1	Optical and Wireless Communication, Signals and Systems

Course Objectives:

- To introduce Python programming and machine learning techniques for wireless communication applications.
- To develop an understanding of modern wireless communication systems such as OFDM, MIMO, and OTFS through simulation.
- To apply machine learning and deep learning models for signal processing, estimation, and detection in wireless systems.
- To familiarize students with AI-enabled physical layer techniques aligned with 3GPP standards.
- To enable students to design and analyze intelligent wireless communication systems using data-driven approaches.

Course Outcomes:

After completing the course, the students should be able to:

1. Implement Python-based simulations for basic and advanced wireless communication systems.
2. Apply machine learning and deep learning techniques for classification, estimation, and detection of wireless signals.
3. Analyze the performance of OFDM, MIMO, and OTFS systems under realistic channel conditions.
4. Design AI/ML-based solutions for channel estimation, modulation classification, spectrum sensing, and NOMA detection.
5. Develop intelligent wireless communication models aligned with 3GPP standards such as CSI feedback, beamforming, and autoencoder-based PHY design.

Unit – I **(11 hours)**

Introduction to Python and Machine Learning Fundamentals: Basics of Python programming: data types, control structures, functions, libraries, Python scientific stack: NumPy, SciPy, Matplotlib, Introduction to Python for signal processing and wireless simulations, Introduction to Machine Learning, Learning paradigms: supervised, semi-supervised, and unsupervised learning, Regression models and classification overview, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Introduction to neural networks and learning concepts

Unit – II **(12 hours)**

Deep Learning Techniques for Wireless Applications: Artificial Neural Networks (ANN), Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM), Generative Adversarial Networks (GANs), Transfer learning concepts and applications, Reinforcement Learning (RL) fundamentals, Use of deep learning frameworks (TensorFlow / PyTorch – conceptual overview)

Unit – III **(11 hours)**
Wireless Communication Systems and Python-Based Modeling: Introduction to wireless communication systems, Single-carrier transmission systems, Orthogonal Frequency Division Multiplexing (OFDM), Multiple-Input Multiple-Output (MIMO) systems, Orthogonal Time Frequency Space (OTFS) modulation, Python-based simulation of single-carrier, OFDM, MIMO, and OTFS systems, Channel models and noise modeling.

Unit – IV **(11 hours)**
AI/ML-Enabled Wireless Signal Processing and Detection: Source coding and channel coding techniques, Modulation and wireless signal classification, Channel estimation techniques, Autoencoder-based wireless signal classification (3GPP-aligned concepts), CSI compression and feedback using autoencoders (3GPP standards), Beamforming and beam management using AI (3GPP perspective), Autoencoder–decoder architectures for PAPR reduction, Spectrum sensing using ML techniques, Successive interference cancellation for NOMA systems, AI/ML-based signal estimation and detection:
Parameter estimation and IF estimation, MIMO/OFDM/OTFS detectors, Denoising of wireless signals

Essential Readings

1. Wireless Communications by A. Goldsmith (Cambridge University Press)
2. Neural Networks and Learning Machines by S. Haykin (Pearson)
3. Deep Learning by Goodfellow *et al.* (MIT Press)

Suggested Readings

1. Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond by T. S. Rappaport *et al.* (IEEE Access)
2. An Introduction to Deep Learning for the Physical Layer by O'Shea and Hoydis (IEEE Transactions on Cognitive Communications)
3. Python documentation and open-source wireless ML frameworks.

<u>List of Experiments</u> (Software based – at least 5) **(30 hours)**

1. Python-based signal generation, noise modeling, and spectral analysis for wireless communication.
2. Simulation of a single-carrier wireless communication system and BER analysis over AWGN and fading channels.
3. Design and performance evaluation of an OFDM system using Python.
4. Simulation and analysis of MIMO wireless systems with linear detection techniques.
5. Modeling and performance comparison of OTFS and OFDM systems in high-mobility channels.
6. Machine learning–based modulation classification using KNN/SVM/CNN.
7. Deep learning–based channel estimation for OFDM/MIMO wireless systems.
8. Autoencoder-based end-to-end wireless communication, CSI compression, and PAPR reduction.
9. ML-based spectrum sensing and successive interference cancellation for NOMA systems.
10. AI/ML-based signal estimation, detection, and denoising including STO, CFO, and symbol rate estimation.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Low Power VLSI Design (DSE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Low Power VLSI Design	4	3	0	1	Concepts of Signals and Systems, Digital Electronics – I

Course Objectives:

- To introduce power/energy fundamentals and the need for low-power CMOS VLSI design.
- To understand sources of power dissipation and key design parameters affecting power.
- To learn a practical low-power design flow and basic power estimation methods from circuit to architecture level.
- To study essential low-power techniques at circuit, logic, and architecture/system levels.
- To expose students to challenges in low power analog circuits and related tecniques.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain dynamic and static power dissipation mechanisms in CMOS and the parameters that influence them.
2. Perform basic power estimation using simulation/activity and interpret power reports at different abstraction levels.
3. Apply circuit- and logic-level low-power techniques such as multi-VDD, sizing, multi-Vt, gating, and encoding.
4. Relate architecture choices to switching activity and overall energy efficiency.
5. Describe standby and runtime power management techniques such as clock/power gating and DVFS.
6. Understand the issues and possible solutions for power management in analog circuits.

Unit – I **(12 hours)**

Introduction to IC design: Basics of analog and digital circuits, design challenges in nano scale CMOS technology. Need for low power VLSI chips; Power vs energy and energy-delay; Power dissipation; Charging and Discharging Capacitance, Short-circuit Current in CMOS Circuit, CMOS Leakage Current, Static Power & Dynamic Power, Parameters for low power design (VDD, capacitance, frequency, VTH, temperature); Design approaches (device / circuit / logic / architecture); Device and technology impact on low power.

Unit – II **(13 hours)**

Low Power architecture: Supply voltage scaling, Device Feature Size Scaling, Architectural-Level Approaches, Voltage Scaling Using High-Level Transformations, Multilevel Voltage Scaling, Dynamic Voltage and Frequency Scaling, Adaptive Voltage Scaling, Subthreshold Logic Circuits, clock gating and clock management, Pipelining and parallelization to reduce supply voltage, Transistor and Gate Sizing, Logical Effort Theory.

Unit – III **(10 hours)**

Low Power Techniques: power consumption in circuits; Dynamic power optimization (multi-V_{DD} concept) and transistor sizing; Static power optimization (multi-V_T concept); Low-power sequential elements (flip-flop / latch power behavior); Logic-level techniques: gate reorganization to reduce glitches.

Unit – IV **(10 hours)**

Low power Analog integrated circuits: challenges in low voltage analog circuit design, Issues about low power supply voltage, roadmap, design of analog circuits using low voltage/low power implementation techniques such as Body bias, Bulk driven, self cascode structure, flipped voltage follower, FG, QFGMOS, DTMOS etc. Weak inversion operation

Suggested Readings

1. Low-Power VLSI Circuits and Systems by Ajit Pal (Springer)
2. Low Power Design Methodologies by J. M. Rabaey and M. Pedram (Springer)
3. Practical Low Power Digital VLSI Design by Gary K. Yeap (Springer)

Suggested Readings

1. Digital Integrated Circuits: A Design Perspective by J. M. Rabaey, A. P. Chandrakasan and B. Nikolic (PH/Pearson)
2. Low-Power CMOS VLSI Circuit Design by K. Roy and S. C. Prasad (Wiley)
3. Low-Power CMOS Design by P. Chandrakasan and R. W. Brodersen (IEEE Press)
4. Leakage in Nanometer CMOS Technologies by Siva Narendra and Anantha Chandrakasan (Springer)
5. CMOS VLSI Design: A Circuits and Systems Perspective by Neil H. E. Weste and David Harris (Pearson)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Estimate dynamic and leakage power of basic CMOS logic across voltage and temperature conditions.
2. Analyze power in small combinational blocks and identify dominant switching and capacitive nodes.
3. Evaluate clock gating techniques and quantify power savings on sequential logic.
4. Explore multi-voltage domain concepts and assess impact on power and design complexity.
5. Compare threshold-voltage options to study leakage vs performance trade-offs.
6. Apply techniques to reduce glitch / switching power through logic-level optimization.
7. Compare FSM implementation choices and observe switching activity and power differences.
8. Study clock distribution power trends using simple clock-tree modeling approaches.
9. Design and compare the power consumption in a bulk-driven amplifier.
10. Study of power consumption in FGMOS/QFGMOS based current mirror.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Fourth Year

1. **Minor in ECE (Offered only to CSE and EE)**
    a) DSE - 6/ GE - 7 : Applied MIMO and B5G Physical Layer Design
    b) DSE - 7/ GE - 8 : Sensors and Actuators for Smart Systems
    c) DSE - 9/ GE - 9 : Statistical Signal Processing
    Or
    Speech and Video Processing
    d) DSE - 10/ GE - 10: Software Defined Radio and Waveform Engineering
    Or
    Compressive Sensing and Sparse Representation

2. **Minor/ Specialization in Telecommunication Networks (Offered to ECE, CSE, and EE)**
    a) DSE - 6/ GE - 7 : Photonic and Optical Communication Networks
    b) DSE - 7/ GE - 8 : Optimization for Engineering
    c) DSE - 9/ GE - 9 : Wireless Systems Implementation
    Or
    Smart Antennas, Beamforming and OTA Testing for 5G/6G
    d) DSE - 10/ GE - 10: Enabling Technologies for 6G Communication
    Or
    Estimation for Wireless Communication

3. **Minor/ Specialization in VLSI Technology and System Design (Offered to ECE, CSE, and EE)**
    a) DSE - 6/ GE - 7 : Microwave Integrated Circuits
    b) DSE - 7/ GE - 8 : Analog VLSI Filter Design
    c) DSE - 9/ GE - 9 : Introduction to Nonlinear Circuits
    Or
    RF Integrated Circuit Design
    d) DSE - 10/ GE - 10: Semiconductor Device Characterization
    Or
    Compact Modeling with Verilog – A

4. **Minor/ Specialization in IoT System Design (Offered to ECE, CSE, and EE)**
    a) DSE - 6/ GE - 7 : Edge IoT Prototyping with RTOS
    b) DSE - 7/ GE - 8 : Industry 4.0 and IIoT
    c) DSE - 9/ GE - 9 : IoT Sensor and Data Integration
    Or
    IoT Reliability Engineering
    d) DSE - 10/ GE - 10: LPWAN IoT Networks
    Or
    Edge Computing

5. Minor/ Specialization in Computer Vision (Offered to ECE, CSE, and EE)

a) DSE - 6/ GE - 7 : Medical Image Processing
b) DSE - 7/ GE - 8 : Remote Sensing and Satellite Image Processing
c) DSE - 9/ GE - 9 : Forensic Image Processing
Or
Computer Graphics
d) DSE - 10/ GE - 10: Computational Photography and Mobile Imaging
Or
Image Restoration and Inpainting

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE)/ Discipline Specific Elective (DSE) courses offered for Minors/ Specializations by the Department in Semester VII

Applied MIMO and B5G Physical Layer Design (DSE – 6 / GE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Applied MIMO and B5G Physical Layer Design	4	3	0	1	Concepts of Digital Communication Systems

Course Objectives:
- Provides in-depth knowledge of MIMO systems and their importance in Beyond-5G (B5G) physical layer design.
- Teaches advanced MIMO signal processing techniques.
- Covers MIMO channel modeling.
- Includes waveform design for next-generation wireless systems.
- Focuses on beamforming methods for improved performance.
- Introduces emerging PHY-layer techniques for B5G.
- Targets systems enabling: High data rates, Low latency, High reliability in wireless communication.

Course Outcomes:
After completing the course, the students should be able to:
1. Analyze and design MIMO wireless communication systems.
2. Apply advanced beamforming and precoding techniques for B5G systems.
3. Understand PHY-layer challenges in B5G such as high mobility, massive connectivity, and spectrum efficiency.
4. Design and evaluate B5G-enabled waveforms and signal processing techniques.
5. Implement and simulate advanced MIMO and B5G physical layer solutions.

Unit – I **(11 hours)**
Fundamentals of MIMO Systems: Review of wireless channel characteristics and fading, SISO, SIMO, MISO, and MIMO system models, Capacity of MIMO channels, Spatial multiplexing and diversity gains, MIMO channel modeling and correlation, Linear MIMO detection techniques: ZF, MMSE, Introduction to space–time coding (Alamouti scheme).

Unit – II **(12 hours)**
Advanced MIMO and Massive MIMO Techniques: Space–time block codes (STBC) and space–time trellis codes (STTC), Non-linear MIMO detection: SIC, Sphere decoding, Precoding techniques: Eigen-beamforming, linear precoding, Massive MIMO systems and scaling laws, Channel estimation and pilot contamination in massive MIMO, Hardware impairments and practical challenge.

Unit – III **(11 hours)**

B5G Physical Layer Waveforms and Channel Models: Limitations of 5G PHY and motivation for B5G, mmWave and Terahertz communication fundamentals, High-mobility and high-frequency channel models, B5G waveforms: OFDM enhancements, Filter bank multicarrier (FBMC), Universal filtered multicarrier (UFMC), Orthogonal Time Frequency Space (OTFS), Performance comparison of B5G waveforms.

Unit – IV **(11 hours)**

Intelligent PHY and B5G Enabling Technologies: Hybrid and digital beamforming for B5G systems, Beam management and beam tracking, Reconfigurable Intelligent Surfaces (RIS) for PHY enhancement, AI/ML-enabled PHY layer design, Joint communication and sensing (ISAC), Ultra-reliable low-latency communication (URLLC) and massive machine-type communication (mMTC), Future directions toward 6G physical layer.

Essential Readings

1. Fundamentals of Wireless Communication by D. Tse and P. Viswanath (Cambridge University Press)
2. Foundations of MIMO Communication by R. Heath and A. Lozano (Cambridge University Press)
3. MIMO-OFDM Wireless Communications with MATLAB by Y. S. Cho *et al.* (Wiley)

Suggested Readings

1. Wireless Communications by A. Goldsmith (Cambridge University Press)
2. Massive MIMO Networks by E. Björnson *et al.* (NOW Publishers)
3. Millimeter Wave Wireless Communications by T. S. Rappaport, R. W. Heath Jr., R. C. Daniels, and J. N. Murdock (Pearson)
4. Fundamentals of Massive MIMO by Thomas L. Marzetta, Erik G. Larsson, Hong Yang, Hien Quoc Ngo (Cambridge University Press)

List of Experiments (Hardware and Software using Matlab/Simulink – at least 5) **(30 hours)**

1. Simulation and capacity analysis of SISO, SIMO, MISO, and MIMO wireless systems.
2. Performance evaluation of spatial diversity and spatial multiplexing in MIMO channels.
3. Implementation and comparison of linear MIMO detectors (ZF and MMSE).
4. Simulation of non-linear MIMO detection techniques including SIC and sphere decoding.
5. Design and performance analysis of space–time block coding (Alamouti scheme).
6. Simulation of massive MIMO systems and study of pilot contamination effects.
7. Performance analysis of beamforming and precoding techniques for MIMO systems.
8. Implementation and comparison of B5G waveforms (OFDM vs FBMC/UFMC/OTFS).
9. Simulation of mmWave/THz channel models and hybrid beamforming techniques.
10. AI/ML-assisted PHY-layer design for MIMO/B5G systems (e.g., channel estimation or detection).

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Sensors and Actuators for Smart Systems (DSE – 7 / GE – 8)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Sensors and Actuators for Smart Systems	4	3	0	1	Concepts of Electrical and Electronics Engineering

Course Objectives:

- To introduce transducers, sensors, and actuators with emphasis on real-world selection and performance parameters.
- To provide working understanding of microfabrication flow and how it relates to microsensors.
- To expose students to important sensor classes including gas sensors, and force / pressure / strain microsensors.
- To build skill in simulation using Multiphysics tool for design / optimization / characterization.
- To develop capability in sensor interfacing.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain operating principles of common sensors/actuators and identify key performance parameters used in evaluation and selection.
2. Describe the microfabrication process flow for microsensors.
3. Use Multiphysics tool to model and optimize a sensor / actuator.
4. Compare gas - sensing approaches and relate choice to application constraints.
5. Interface sensors and implement basic filtering, and perform calibration / characterization.

Unit – I **(11 hours)**

Transducers, Sensors & Performance Parameters: transducers; sensor / actuator classification; static and dynamic characteristics; calibration. Overview of practical sensing mechanisms - resistive / capacitive / piezoelectric / optical / magnetic / MEMS.

Unit – II **(12 hours)**

Microtechnology & Sensor Realization: Thin-film physics, thin film deposition (CVD / PVD), photolithography and patterning, etching methods, MEMS / microsensor structures. Microsensors: force / pressure / strain gauge.

Unit – III **(11 hours)**

Sensor Classes: Gas sensors – optical / MOS / FET, piezoelectric; selection criteria (selectivity, sensitivity, response time, drift). Basics of microfluidics. Actuators: operating principles of piezo / piezoresistive and electrostatic actuation; micropumps and micro-actuators.

Unit – IV **(11 hours)**

Simulation and Characterization: Simulation (Multiphysics), sensitivity, dynamic response; optimization and characterization using EDA. Sensor interfacing with microprocessor / microcontroller.

Essential Readings

1. Handbook of Modern Sensors: Physics, Designs, and Applications by J. Fraden (Springer)
2. An Introduction to the Finite Element Method by J.N. Reddy (McGraw-Hill Education)
3. Advances in Multiphysics Simulation and Experimental Testing of MEMS by Attilio Frangi (Springer)
4. Fundamentals of Microfabrication and Nanotechnology by Marc J. Madou (CRC Press)

Suggested Readings

1. Modeling MEMS and NEMS by Peleshko & Bernstein (CRC Press)
2. The Art of Electronics by Paul Horowitz and Winfield Hill (Cambridge University Press)
3. COMSOL Multiphysics Documentation / Application Library by COMSOL AB (Official Documentation)
4. Microsystem Design by Stephen D. Senturia (Springer)

List of Experiments (Hardware and Software – at least 5) **(30 hours)**

1. To acquire sensor data and determine basic static characteristics with calibration.
2. To control an actuator and implement safe operating limits and stable control logic.
3. To model a mechanical sensor (Multiphysics Simulation) structure and extract sensitivity from simulation results.
4. To simulate a micro actuator principle in a Multiphysics setup.
5. To simulate thermal behaviour and evaluate time constant and sensitivity under heating/cooling.
6. To simulate diffusion/transport behaviour relevant to gas sensing.
7. To model basic microfluidic flow behaviour.
8. To integrate sensors and actuators into a subsystem.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Photonic and Optical Communication Networks (DSE – 6 / GE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Photonic and Optical Communication Networks	4	3	0	1	NIL

Course Objectives:

- Explain optical transmission techniques for high-speed data transport.
- Develop understanding of the design and operation of modern optical communication networks.
- Emphasize wavelength-based networking concepts (WDM-based systems).
- Introduce optical switching techniques and their role in networking.
- Study different optical network architectures and design principles.
- Explore emerging photonic technologies enabling high-capacity communication systems.

Course Outcomes:
After completing the course, the students should be able to:

1. Understand the fundamentals of photonic and optical communication systems.
2. Analyze optical transmission impairments and system performance.
3. Design wavelength-routed optical networks and optical switching systems.
4. Understand optical network architectures and control mechanisms.
5. Evaluate emerging photonic technologies for future optical networks.

Unit – I **(11 hours)**
Fundamentals of Photonic and Optical Communication: Review of optical fiber communication principles, Optical transmission windows and fiber types, Optical sources and detectors (laser diodes, photodiodes), Optical amplifiers: EDFA, Raman amplifiers, Fiber impairments: attenuation, dispersion, nonlinear effects, System performance parameters and link design basics.

Unit – II **(11 hours)**
Wavelength Division Multiplexing and Optical Switching: Principles of WDM and DWDM systems, Optical multiplexers and demultiplexers, Optical add-drop multiplexers (OADM) and reconfigurable OADMs (ROADM), Optical switching techniques: space, wavelength, and time switching, Optical cross-connects (OXC), Protection and restoration in optical networks.

Unit – III **(12 hours)**
Optical Network Architectures and Control: Optical access networks: PON, FTTH, FTTx, Metropolitan and long-haul optical networks, Wavelength-routed networks and routing and wavelength assignment (RWA), Optical packet switching and optical burst switching, Control and management of optical networks, GMPLS and SDN-enabled optical networks.

Unit – IV **(11 hours)**
Advanced and Emerging Photonic Networks: Elastic optical networks (EON), Coherent optical communication systems, Space-division multiplexing (SDM) and multi-core fibers, Optical wireless and free-space optical (FSO) networks, Photonic integration and silicon photonics, Future trends toward B5G/6G optical networking.

Essential Readings

1. Optical Fiber Communications by G. Keiser (McGraw-Hill)
2. Optical Networks: A Practical Perspective by R. Ramaswami, K. Sivarajan, and G. Sasaki (Morgan Kaufmann)
3. Optical Fiber Communications: Principles and Practice by J. M. Senior and M. Jamro (Pearson)

Suggested Readings

1. Optical Communication Networks by B. Mukherjee (McGraw-Hill)
2. Optical Fiber Telecommunications by I. Kaminow *et al.* (Academic Press)
3. Fiber-Optic Communication Systems by G. P. Agrawal (Wiley)

List of Experiments (Hardware / OptiSystem / OptiSim – at least 5) **(30 hours)**

1. Simulation and analysis of an optical fiber link including attenuation and dispersion effects.
2. Design and performance evaluation of optical transmitters using LED and laser diode sources.
3. Performance analysis of PIN and APD photodiode–based optical receivers.
4. Simulation of optical amplification using EDFA and Raman amplifiers.
5. Design and analysis of WDM/DWDM optical communication systems.
6. Performance evaluation of optical add-drop multiplexers (OADM) and ROADM networks.
7. Simulation of wavelength-routed optical networks and RWA algorithms.
8. Study of protection and restoration schemes in optical networks.
9. Simulation and performance analysis of coherent optical communication systems.
10. Design and evaluation of free-space optical (FSO) communication links under atmospheric effects.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Optimization for Engineering (DSE – 7 / GE – 8)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Optimization for Engineering	4	3	0	1	Concepts of Mathematics - I

Course Objectives:
- To introduce fundamental concepts of numerical optimization.
- To study unconstrained and constrained optimization techniques.
- To understand gradient-based and iterative optimization algorithms.
- To apply numerical optimization methods to engineering problems.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain principles of numerical optimization.
2. Solve unconstrained optimization problems using numerical methods.
3. Apply constrained optimization techniques.
4. Implement optimization algorithms for real-world engineering applications.

Unit – I **(12 hours)**
Basics of Optimization: Optimization problems and applications, classification of optimization problems, convexity, optimality conditions, numerical issues and convergence.

Unit – II **(11 hours)**
Unconstrained Optimization Methods: One-dimensional search methods, gradient descent, steepest descent, Newton's method, quasi-Newton methods, convergence analysis.

Unit – III **(11 hours)**
Constrained Optimization: Equality and inequality constraints, Lagrange multipliers, Karush–Kuhn–Tucker (KKT) conditions, penalty and barrier methods.

Unit – IV **(11 hours)**
Advanced and Practical Optimization Techniques: Nonlinear programming, least squares optimization, stochastic and heuristic methods, applications in signal processing and machine learning.

Essential Readings
1. An Introduction to Optimization by Edwin K. P. Chong and Stanislaw H. Żak (Wiley).
2. Convex Optimization by S. Boyd and L. Vandenberghe (Cambridge University Press)
3. Optimization for Engineering Design by K. Deb (PHI)

Suggested Readings
1. Numerical Optimization by J. Nocedal and S. Wright (Springer)
2. Nonlinear Programming by D. P. Bertsekas (Athena Scientific)

<u>List of Experiments</u> (Hardware / Software) **(30 hours)**

1. Implementation of one-dimensional search methods.
2. Gradient descent and Newton's method for optimization.
3. Constrained optimization using Lagrange multipliers.
4. Implementation of penalty and barrier methods.
5. Least squares optimization problems.
6. Case studies pertaining to Communication Networks using numerical optimization.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Microwave Integrated Circuits (DSE – 6 / GE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Microwave Integrated Circuits	4	3	0	1	Concepts of Electromagnetic Theory / Electromagnetic Field Theory

Course Objectives:

- To introduce microwave IC passive structures and S-parameter based design workflows.
- To design and verify low-noise amplifiers using stability, gain and noise design techniques with practical biasing and matching.
- To design power amplifiers focusing on power/efficiency/linearity metrics and practical output matching (load-pull).
- To design RF switches and integrate LNA–PA–Switch.

Course Outcomes:

After completing the course, the students should be able to:

1. Design microwave passive networks and evaluate performance using S-parameters.
2. Design an LNA and verify stability, gain, noise figure and linearity targets through simulation and layout-aware checks.
3. Design a PA and evaluate P_{sat} / P_{1dB}/IP_3 and efficiency (PAE) with practical matching and stability considerations.
4. Design RF switches and analyze insertion loss, isolation, linearity and power handling trade-offs.
5. Integrate and layout a T/R front-end.

Unit – I **(10 hours)**

Building Blocks: Microstrip/CPW transmission lines, matching, impedance transformation and matching networks, stubs and quarter-wave sections, discontinuities, couplers and power dividers, resonators and basic filter blocks, S-parameters, insertion/return loss.

Unit – II **(12 hours)**

Active Devices and LNA Design: Small-signal models at microwave frequencies, stability circles, gain circles and noise circles, matching for LNA, noise figure, gain trade-offs, linearity metrics, single-stage and two-stage LNA design.

Unit – III **(13 hours)**

Power Amplifier (PA) Design: PA classes, power / compression / efficiency metrics (P_{1dB}, P_{sat}, PAE), load-pull concept and output matching, harmonic, stability under large signal, distortion, thermal / EM considerations.

Unit – IV **(10 hours)**

Switches and T/R: RF switch fundamentals, SPDT switch, T/R switch, LNA + PA + switch + matching, isolation, parasitic management, EM co-simulation, and layout fundamentals.

Essential Readings

1. Microwave Engineering by David M. Pozar (Wiley)
2. Microwave Transistor Amplifiers: Analysis and Design by Guillermo Gonzalez (Prentice Hall)
3. The Design of CMOS Radio-Frequency Integrated Circuits by Thomas H. Lee (Cambridge University Press)
4. RF Power Amplifiers for Wireless Communications by Steve C. Cripps (Artech House)

Suggested Readings

1. RF Microelectronics by Behzad Razavi (Prentice Hall)
2. Microstrip Filters for RF / Microwave Applications by Jia-Sheng Hong and M. J. Lancaster (Wiley)
3. RF Circuit Design: Theory and Applications by Reinhold Ludwig and Pavel Bretchko (Pearson)
4. RF / Microwave Circuit Design for Wireless Applications by Ulrich L. Rohde and David P. Newkirk (Wiley).

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Model transmission lines and basic discontinuities to extract impedance and loss trends across frequency.
2. Design and verify impedance matching networks to meet a target return loss over a given band.
3. Design a passive RF network block and validate key S-parameter performance.
4. Design a basic RF amplifier stage with input/output matching to meet gain and stability goals.
5. Perform stability checks and apply stabilization methods while tracking performance impact.
6. Evaluate amplifier linearity and trade-offs using standard RF metrics and simple design tweaks.
7. Design a basic power stage with output matching and study output power/efficiency behavior.
8. Integrate multiple RF blocks into a simple front-end chain and verify end-to-end performance.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Analog VLSI Filter Design (DSE – 7 / GE – 8)

(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Analog VLSI Filter Design	4	3	0	1	Concepts of Electronic Devices and Circuits / Analog and Digital Electronic Circuits / CMOS Analog IC Design / Current Mode Analog VLSI Circuits

Course Objectives:

- To understand the various filter approximations methods, the frequency scaling, impedance scaling and the various filter transformations.
- To understand the various standard analog filter design approaches.
- To be able to design second order and higher order voltage-mode and current-mode analog filters modern analog electronic circuit building blocks.
- To be able to design fully-integratable Gm-C and OTA-C Filters, MOS Switched-Capacitor Filters and continuous-time MOSFET-C filters.
- To understand the design of log-domain / translinear current-mode (CM) filters.
- To understand the design of CMOS CM square-root domain (SQRD) filters.

Course Outcomes:

After completing the course, the students should be able to:

1. Determine suitable filter function to match the given filter specifications and devise a prototype design.
2. Decide the suitable technology and configuration to be used to design the required filter.
3. Design Gm-C or OTA-C topology and evaluate its performance analytically and experimentally through SPICE simulations.
4. Design a fully-integrable filter using modern analogue circuit building blocks.
5. Design of CMOS compatible Switched-Capacitor or MOSFET-C Filter and justify the choice by using various features.
6. Design a translinear or square root domain filter and verify the design using SPICE simulations

Unit – I **(15 hours)**

Standard analog filter functions, responses; passive RC and RLC filters: first order, second order and higher order; filter approximations: Butterworth, Chebyshev, Inverse Chebyshev; frequency normalization and demoralization; frequency-scaling; impedance-scaling; transformations: LP to HP, LP to BP and LP to BS.

Unit – II **(10 hours)**

Analog filter design: state-variable synthesis of biquad filters, Simulated inductor and FDNR, Filter design using GIC; Leapfrog active ladder filter synthesis to higher order filter design; design of active R inductors, resonators and biquad active filters. CMOS OTAs, transconductors; component simulators using Gm-C and OTA-C, Gm-C inductors, FDNRs; Second order VM and CM biquad designs; higher order Gm-C filter designs.

Unit – III **(10 hours)**

The Current Conveyors, Mixed Translinear Cells (MTL), Translinear realizations of CCII-, CCI+/- and the Controlled Current Conveyor II, Bipolar / CMOS implementations of CCII / CFOAs; internal circuit architectures of modern analog circuit building blocks, Exemplary Universal Voltage-mode and Current-mode filter designs using modern building blocks.

Unit – IV **(10 hours)**

MOS Switched-Capacitor (SC) Filters; SC biquads, higher order SC Filters using bilinear transformation; Techniques of Nonlinearity-cancellation in MOS analog Circuits, MOS Resistive Circuits (MRC); continuous-time MOSFET-C integrators and MOSFET-C filters using CCII and CFOAs; fully-integratable log-domain translinear and CMOS square-root domain (SQRD) filters.

Essential Readings

1. 1. Current-mode VLSI Analog Filters: Design and Applications by P. V. Ananda Mohan (Birkhauser-Boston)
2. Analog Filter Design by M. E. Van Valkenburg and Rolf Schaumann (Holt-Saunders International Edition)
3. Design of Analog Filters: Passive, Active RC and Switched Capacitor by Kenneth Laker, M. S. Ghausi, and Rolf Schumann (Prentice Hall)

Suggested Readings

1. Active and Passive Analog Filter Design: An Introduction by Lawrence P. Huelsman (McGraw-Hill)
2. Active Filter Design by Allan Waters (Macmillan Education)
3. Linear Active Networks: Theory and Design by R. E. Thomas, A. J. Rosa, G. J. Toussaint (Wiley)
4. Analog Filter Design Demystified by Louis E. Frenzel Jr. (McGraw-Hill)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Design of LP, BP and HP Filters using Universal Active Filter (UAF) IC
2. Design of All pass and Notch Filters using Simulated inductors
3. Active-R filter design
4. OTA-C Universal Biquad Filter design
5. Higher order Filter Design using FDNR approach
6. Filter design using active-compensated Tow-Thomas Biquad
7. Switched-Capacitor Filter Design
8. Fully-differential MOSFET-C biquad Filter Design
9. Log-domain and Translinear Filter design
10. Design of a Square-root domain Filter

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Edge IoT Prototyping with RTOS (DSE – 6 / GE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Edge IoT Prototyping with RTOS	4	3	0	1	Introduction to Embedded Systems for IoT

Course Objectives:

- To develop reliable IoT firmware using RTOS building blocks.
- To implement device connectivity and messaging using Wi-Fi / BLE and application messaging using MQTT / HTTP / CoAP.
- To apply practical low-power techniques using sleep modes and duty cycling.
- To design end-to-end edge prototypes integrating sensors, actuators, telemetry, and remote control.

Course Outcomes:

After completing the course, the students should be able to:

1. Develop RTOS-based firmware with multi-task scheduling, IPC, and timing guarantees.
2. Interface sensors and actuators using standard peripherals and build reusable driver modules.
3. Implement robust telemetry and command / control over Wi-Fi / BLE using MQTT / HTTP / CoAP with retries and offline handling.
4. Measure and reduce power consumption using sleep states, wake sources, duty cycling, and current profiling methods.

Unit – I **(12 hours)**

RTOS concepts: tasks / threads, priorities, scheduling and preemption, context switching, semaphores and mutex, message queues, event flags, ISR vs task context, exceptions, interrupts, timers and timer services.

Unit – II **(11 hours)**

Interfacing: interrupts, I2C / SPI / UART basics, ADC sampling patterns and timer-triggered acquisition, debouncing, digital filtering, PWM actuation, storing configuration (EEPROM / Flash / NVS).

Unit – III **(12 hours)**

Connectivity and Messaging: Wi-Fi provisioning and reconnect patterns, BLE GATT services for configuration / telemetry, MQTT publish / subscribe patterns, HTTP / REST and CoAP basics, payload formats (JSON / CBOR).

Unit – IV **(10 hours)**

Low Power and Integration: firmware update concept, rollback, doing more with less (RAM / code / cycles), reducing power consumption, reliability patterns (watchdog), diagnostics (health metrics, last-seen).

Essential Readings

1. Making Embedded Systems: Design Patterns for Great Software by Elecia White (O'Reilly Media)
2. Real-Time Concepts for Embedded Systems by Qing Li and Caroline Yao (CRC Press)
3. Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti (Universities Press)
4. MQTT Essentials by HiveMQ (HiveMQ)

Suggested Readings

1. Designing Connected Products by Claire Rowland, Elizabeth Goodman, Martin Charlier, Alfred Lui, and Ann Light (O'Reilly Media)
2. Embedded Software Design: A Practical Approach to Architecture, Processes, and Coding Techniques by Jacob Beningo (Elsevier)
3. Advanced PIC Microcontroller Projects in C by Dogan Ibrahim (Newnes)
4. FreeRTOS Documentation, FreeRTOS.org (official docs).

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Develop a basic RTOS based firmware application.
2. Implement event driven input handling using GPIO interrupts.
3. Interface a digital sensor and convert raw data into usable physical values with periodic sampling.
4. Acquire and process analog signals using ADC with simple filtering.
5. Control an actuator output (PWM/GPIO) with timeout / limit behavior.
6. Publish device telemetry over a lightweight messaging protocol.
7. Enable remote control features with command reception and status / acknowledgement responses.
8. Add reliability and power features such as offline buffering/retry and basic low-power duty cycling.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Industry 4.0 and IIoT (DSE – 7 / GE – 8)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Industry 4.0 and IIoT	4	3	0	1	Introduction to Embedded Systems for IoT

Course Objectives:

- To introduce the Industry 4.0 vision and how IIoT enables smart factories and smart operations.
- To introduce core building blocks: sensing / actuation, industrial connectivity, and gateways.
- To develop practical understanding of industrial data flow.
- To introduce basic analytics such as monitoring, anomaly detection, predictive maintenance.
- To build awareness of cybersecurity, safety, and reliability requirements in industrial environments.

Course Outcomes:

After completing the course, the students should be able to:

1. Describe Industry 4.0 concepts and identify where IIoT fits in manufacturing automation.
2. Explain an IIoT system architecture using layers.
3. Select suitable industrial connectivity and messaging approaches for common industrial scenarios.
4. Build a basic IIoT data pipeline prototype and interpret key metrics.
5. Apply basic security and reliability practices (identity, authentication, segmentation) for industrial deployments.

Unit – I **(10 hours)**

Industry 4.0 overview: evolution of industrial revolutions, smart factory idea. IIoT fundamentals: sensing & actuation, controllers and gateways, basic communication and networking, data lifecycle. Industrial constraints (uptime, latency, safety, cost).

Unit – II **(12 hours)**

IIoT Architecture: device layer, edge / gateway layer, network layer, platform layer, application layer. Connectivity Options: wired vs wireless, reliability, edge gateway roles. Industrial Protocols: MQTT for telemetry, Modbus basics, REST APIs; integration with SCADA / MES / ERP.

Unit – III **(12 hours)**

Edge vs fog vs cloud roles, local processing and buffering, device management concepts, basic streaming. Industrial data: time-series, event logs, alarms, KPIs. Analytics: descriptive monitoring, simple anomaly detection, and alerting.

Unit – IV **(11 hours)**

Security for IIoT: confidentiality, integrity, availability, safety; identity and authentication, secure communication, access control, network segmentation, logging / monitoring. Reliability: fail-safe behavior, redundancy intuition.

Essential Readings

1. Introduction to Industrial Internet of Things and Industry 4.0 by Sudip Misra, Chandana Roy, and Anandarup Mukherjee (CRC Press)
2. Sensors, Cloud, and Fog: The Enabling Technologies for the Internet of Things by Sudip Misra, Subhadeep Sarkar, Subarna Chatterjee (CRC Press)
3. Industry 4.0: The Industrial Internet of Things by Alasdair Gilchrist (Apress)
4. Practical Industrial Cybersecurity: ICS, Industry 4.0, and IIoT by Charles J. Brooks, Philip A. Craig Jr. (Wiley)

Suggested Readings

1. Introduction to IoT by Sudip Misra, Anandarup Mukherjee, and Arijit Roy (Cambridge University Press)
2. Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS by Tyson Macaulay and Bryan L. Singer (CRC Press)
3. Designing Connected Products by Claire Rowland, Elizabeth Goodman, Martin Charlier, Alfred Lui, and Ann Light (O'Reilly Media)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Acquire or simulate sensor data and produce a clean, timestamped time series stream.
2. Demonstrate a publish / subscribe data flow for telemetry and basic command/control using a simple message format.
3. Model an industrial device communication interface.
4. Implement basic edge processing such as filtering, aggregation, and rate reduction before upstream publishing.
5. Add rule-based alert / event generation and log alarms with timestamps.
6. Run a simple anomaly detection experiment and evaluate detection vs false-alarm behavior.
7. Compare edge vs cloud workflows using basic latency and bandwidth measurements under different streaming modes.
8. Apply basic security controls such as encrypted communication and access control concepts/configuration.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Medical Image Processing (DSE – 6 / GE – 7)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Medical Image Processing	4	3	0	1	Fundamentals of Image Processing, Image Filtering and Restoration

Course Objectives:

- Develop the ability to handle medical image data formats, metadata, and anonymization for safe and reproducible analysis workflows.
- Design and implement medical-image-specific preprocessing and enhancement.
- Build and evaluate segmentation and detection using appropriate clinical metrics and error analysis.
- Perform registration and quantitative measurement extraction.

Course Outcomes:

After completing the course, the students should be able to:

1. Load, visualize and preprocess DICOM / NIfTI and multi-modal datasets.
2. Implement and compare classical and DL segmentation.
3. Perform image registration and evaluate alignment quality.
4. Extract quantitative biomarkers and present results.
5. Design a small end-to-end medical imaging prototype.

Unit – I **(12 hours)**

Medical Imaging Basics: Imaging modalities (X-ray, CT, MRI, Ultrasound). Data formats: DICOM tags, NIfTI volumes, slice spacing, windowing / leveling, anonymization. Dataset preparation: annotation tools, train / val / test splits, class imbalance, augmentation specific to medical images.

Unit – II **(11 hours)**

Preprocessing: Intensity normalization, CLAHE, denoising, deblurring basics, artifact reduction. ROI extraction, body/organ masking, edge-preserving filtering, contrast tuning for X-ray/CT.

Unit – III **(11 hours)**

Segmentation & Detection: Classical segmentation (thresholding, region growing, active contours) and deep segmentation (U-Net / Attention U-Net concepts, patch-based training). Losses (Dice, focal), post-processing (morphology, connected components). Metrics: Dice/IoU, sensitivity/specificity, calibration and confidence visualization.

Unit – IV **(11 hours)**

Registration and 3D Analysis: Rigid / affine registration, similarity measures (SSD / NCC / MI basics), resampling and interpolation. 3D volume processing, extracting measurements (lesion volume, organ size).

Suggested Readings

1. Medical Image Analysis by Atam P. Dhawan (Wiley-IEEE Press)
2. Medical Image Processing, Reconstruction and Restoration: Concepts and Methods by Jiri Jan (CRC Press)
3. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson)
4. Biomedical Image Analysis by Rangaraj M. Rangayyan (CRC Press)

Essential Readings

1. Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis by Terry Yoo (A K Peters/CRC Press)
2. The Image Processing Handbook by John C. Russ (CRC Press)
3. Deep Learning for Medical Image Analysis by S. Kevin Zhou, Hayit Greenspan, and Dinggang Shen (Academic Press/Elsevier)

List of Experiments (Software using Matlab / Python) **(30 hours)**

1. To load and visualize DICOM and NIfTI images to extract key metadata.
2. To anonymize a DICOM study safely to verify removal of identifying tags and generate a hashed patient ID mapping.
3. To implement CT windowing / leveling presets to create consistent visualization outputs for reports.
4. To benchmark denoising methods on X-ray/CT images to compare detail preservation using PSNR / SSIM.
5. To build ROI extraction to reduce background clutter and improve downstream model accuracy.
6. To implement classical segmentation to compare against DL segmentation on the same dataset.
7. To train a U-Net style segmentation model on a small labeled dataset to evaluate Dice/IoU and analyze major failure cases.
8. To compare loss functions to study behavior under class imbalance and small lesion sizes.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Remote Sensing and Satellite Image Processing (DSE – 7 / GE – 8)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Remote Sensing and Satellite Image Processing	4	3	0	1	Fundamentals of Image Processing, Image Filtering and Restoration

Course Objectives:

- Develop practical skills to handle satellite raster data using core GIS concepts such as CRS / projections, geo-referencing, clipping, mosaicking, and map layouts.
- Perform standard multispectral preprocessing and generate features.
- Apply SAR processing and interpret SAR backscatter for mapping applications.
- Design and implement classification/segmentation for land cover and object extraction using ML.

Course Outcomes:

After completing the course, the students should be able to:

1. Prepare analysis-ready satellite datasets by applying GIS operations.
2. Generate and interpret multispectral features and use them for mapping tasks.
3. Process SAR imagery using standard steps and create application-specific SAR-derived maps.
4. Train and evaluate supervised classifiers for land-cover mapping using proper sampling, validation splits, and accuracy metrics.

Unit – I **(12 hours)**

Foundations: Remote sensing fundamentals (EM spectrum, sensors, spatial / spectral / temporal resolution, radiometric resolution). GIS basics for imagery: CRS / projections, geo-referencing, resampling, mosaicking, tiling, and spatial extent / scale.

Unit – II **(11 hours)**

Image Preprocessing and Feature Generation: Radiometric / atmospheric correction, cloud / shadow masking, normalization across dates. Band combinations (true / false color) and feature engineering: indices, textures (GLCM), transforms for land-cover separability. Sampling strategy, training label creation.

Unit – III **(11 hours)**

SAR Image Processing: SAR basics, speckle characteristics and filtering (Lee / Frost / Refined Lee), radiometric calibration (Sigma0 / Gamma0 / Beta0), terrain correction / orthorectification, and multi-temporal compositing.

Unit – IV **(11 hours)**

Assessment: Supervised classification, class imbalance, confusion matrix and map accuracy metrics, area estimation. Change detection: index differencing, image ratioing, post-classification comparison, time-series trend basics.

Essential Readings

1. Remote Sensing Digital Image Analysis: An Introduction by John A. Richards and Xiuping Jia (Springer)
2. Remote Sensing of the Environment: An Earth Resource Perspective by John R. Jensen (Pearson)
3. Remote Sensing and Image Interpretation by Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman (Wiley)

Suggested Readings

1. Remote Sensing: Models and Methods for Image Processing by Robert A. Schowengerdt (Academic Press/Elsevier)
2. Synthetic Aperture Radar: Systems and Signal Processing by John C. Curlander and Robert N. McDonough (Wiley-Interscience)

List of Experiments (Software using Matlab / Python – at least 5) **(30 hours)**

1. Create band composites for visual interpretation and validation using reference layers.
2. Perform basic raster preprocessing and observe impact on output quality.
3. Compute common spectral indices and generate thematic maps using simple thresholds.
4. Generate basic texture or auxiliary features to improve separability for baseline analysis.
5. Run a simple supervised classification and evaluate results using standard accuracy metrics.
6. Perform basic change detection between two dates.
7. Process SAR imagery and compare interpretability.
8. Generate a simple SAR based water / flood map and validate qualitatively using available reference data.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE)/ Discipline Specific Elective (DSE) courses offered for Minors/ Specializations by the Department in Semester VIII

Statistical Signal Processing (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Statistical Signal Processing	4	3	0	1	Concepts of Signals and Systems, Digital Signal Processing

Course Objectives:
- Provide a strong foundation in probability theory and random processes for signal processing.
- Enable understanding of statistical models of signals and noise.
- Introduce estimation and detection theory for practical signal processing problems.
- Develop the ability to design adaptive and statistical signal processing algorithms.
- Analyze system performance under uncertainty using statistical measures.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain probabilistic concepts and random process models used in statistical signal processing.
2. Apply statistical estimation techniques for parameter extraction from noisy signals.
3. Implement detection and hypothesis testing methods for signal processing applications.
4. Analyse adaptive filtering techniques for time-varying signals.

Unit – I **(11 hours)**
Probability Theory and Random Processes: Review of probability concepts; random variables and probability distributions; expectation, moments, covariance; multiple random variables; random processes; stationarity and ergodicity; autocorrelation and power spectral density; noise models in signal processing.

Unit – II **(12 hours)**
Statistical Estimation Theory: Point and interval estimation; bias, variance, and efficiency; minimum mean square error (MMSE) estimation; maximum likelihood (ML) estimation; Bayesian estimation; Cramér–Rao lower bound; applications to signal parameter estimation.

Unit – III **(11 hours)**
Detection and Hypothesis Testing: Binary and multiple hypothesis testing; Neyman - Pearson criterion; likelihood ratio tests; Bayesian detection; matched filter; signal detection in Gaussian noise.

Unit – IV **(11 hours)**

Adaptive and Advanced Statistical Signal Processing: Adaptive filtering concepts; LMS and RLS algorithms; convergence analysis; spectral estimation methods; Kalman filtering basics; applications in communications, radar, biomedical, and audio signal processing.

Essential Readings

1. Statistical Digital Signal Processing and Modeling by Monson H. Hayes
2. Fundamentals of Statistical Signal Processing, Vol. I (Estimation Theory) & Vol. II (Detection Theory) by S. M. Kay (Pearson)
3. Adaptive Filter Theory by S. Haykin (Pearson)

Suggested Readings

1. Probability, Random Variables, and Stochastic Processes by A. Papoulis and S. U. Pillai (McGraw-Hill)
2. Detection, Estimation, and Modulation Theory by H. L. Van Trees (Wiley)
3. An Introduction to Statistical Signal Processing by R. M. Gray and L. D. Davisson (Cambridge University Press)

List of Experiments (Hardware / Software using Matlab / Simulink – at least 5) **(30 hours)**

1. Simulation of random variables and probability distributions.
2. Generation and analysis of random processes.
3. Statistical characterization of noise signals.
4. Parameter estimation using ML and MMSE techniques.
5. Binary hypothesis testing and ROC curve analysis.
6. Matched filter implementation for signal detection.
7. Adaptive filtering using LMS algorithm.
8. Adaptive filtering using RLS algorithm.
9. Spectral estimation of random signals.
10. Mini-project on a statistical signal processing application.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Speech and Video Processing (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Speech and Video Processing	4	3	0	1	Concepts of Signals and Systems

Course Objectives:

- Understand the fundamentals of speech and video signal processing.
- Analyze speech production, perception, and representation techniques.
- Apply digital video processing methods for enhancement and analysis.
- Design algorithms for speech and video coding, recognition, and analysis.
- Evaluate speech and video processing systems for real-world multimedia applications.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain speech and video signal characteristics and representation methods.
2. Apply speech analysis techniques for feature extraction and enhancement.
3. Implement video processing techniques including motion estimation and compression.
4. Analyze speech and video coding and recognition algorithms.
5. Design and evaluate speech and video processing solutions for multimedia applications.

Unit – I **(11 hours)**

Speech production and perception mechanism; speech signal characteristics; sampling and quantization of speech signals; short-time analysis; time-domain and frequency-domain representations; speech databases.

Unit – II **(12 hours)**

Linear predictive coding (LPC); cepstral analysis; Mel-frequency cepstral coefficients (MFCC); speech enhancement techniques; speech coding standards; speech recognition and synthesis overview.

Unit – III **(11 hours)**

Video signal representation; color models and spaces; human visual system; spatial and temporal redundancy; video sampling and formats; basic video enhancement techniques.

Unit – IV **(11 hours)**

Motion estimation and compensation; transform coding; video compression standards (MPEG, H.264/AVC, HEVC); video segmentation and object tracking; applications in surveillance, multimedia, and communication.

Essential Readings

1. Digital Processing of Speech Signals by L. R. Rabiner and R. W. Schafer (Pearson)
2. Discrete-Time Processing of Speech Signals by J. R. Deller, J. H. L. Hansen, and J. G. Proakis (IEEE Press)
3. Digital Video Processing by A. Murat Tekalp (Prentice Hall)

Suggested Readings

1. Multimedia Communication Systems by K. R. Rao, Z. S. Bojkovic, and D. A. Milovanovic (Prentice Hall)
2. Digital Image Processing by R. C. Gonzalez and R. E. Woods (Pearson)

List of Experiments (Hardware / Software using Matlab / Simulink – at least 5) **(30 hours)**

1. Speech signal acquisition and visualization.
2. Time-domain and frequency-domain analysis of speech signals.
3. Speech feature extraction using LPC and MFCC.
4. Speech enhancement using filtering techniques.
5. Speech coding and compression experiment.
6. Video frame extraction and color space conversion.
7. Video enhancement using spatial filtering.
8. Motion estimation and motion compensation.
9. Video compression using MPEG/H.264 standards.
10. Mini-project on speech or video processing application.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Software Defined Radio and Waveform Engineering (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Software Defined Radio and Waveform Engineering	4	3	0	1	Concepts of Signals and Systems / Digital Signal Processing

Course Objectives:

- Introduce SDR fundamentals and I/Q signal concepts.
- Implement basic receivers/transmitters using block-based tools (GNU Radio).
- Learn debugging using spectrum / constellation / time plots.
- Develop practical ability to build basic SDR receiver chains (tuning, filtering, demodulation).
- Train students to handle real-world impairments (noise, CFO, timing offset, interference).

Course Outcomes:

After completing the course, the students should be able to:

1. Configure SDR hardware / software and correctly capture, visualize, and interpret I/Q signals in time and frequency domains.
2. Design and implement basic modulation / demodulation (AM / FM / FSK and BPSK / QPSK) using SDR tools.
3. Apply practical synchronization methods (preamble / correlation, CFO correction) and improve receiver robustness.
4. Evaluate waveform performance using simple KPIs (spectrum, constellation, BER/PER trends) and justify parameter choices.

Unit – I **(10 hours)**

SDR concept and architecture, I/Q representation and complex signals, sampling rate / aliasing, gain staging and dynamic range, time / frequency visualization (FFT), SDR using GNU Radio / MATLAB.

Unit – II **(12 hours)**

Digital mixing and frequency translation, FIR filtering, decimation / interpolation and resampling, windowing and PSD estimation, AGC, DC offset / IQ imbalance.

Unit – III **(12 hours)**

AM / FM receiver chain, FSK / GFSK for telemetry / IoT, BPSK / QPSK (constellation view), pulse shaping (RRC), symbol rate vs sample rate planning, constellation / eye diagram interpretation and tuning knobs.

Unit – IV **(11 hours)**

Noise and SNR, frequency offset and phase noise, timing offset and symbol timing recovery, preamble-based synchronization, equalization, interference and adjacent channel issues, robustness: oversampling, filtering, thresholds, retry framing.

Essential Readings

1. Software Defined Radio using MATLAB & Simulink and the RTL-SDR by Robert W. Stewart (Strathclyde Academic Media)
2. Understanding Digital Signal Processing by Richard G. Lyons (Pearson)
3. Digital Communications: Fundamentals and Applications by Bernard Sklar (Pearson)
4. GNU Radio Tutorial Available at: https://wiki.gnuradio.org/index.php/Tutorials

Suggested Readings

1. Software Defined Radio for Engineers by Travis F. Collings, Robin Getz, Di Pu, and Alexander M. Wyglinski (Artech House)
2. Communication Systems by Simon Haykin (Wiley)
3. Signals and Systems by Alan V. Oppenheim and Alan S. Willsky (Pearson)

List of Experiments (Hardware / Software using Matlab / Simulink – at least 5) **(30 hours)**

1. Set up an SDR receiver and verify basic operations.
2. Capture and replay I/Q samples and compare time/frequency-domain behavior offline.
3. Build a basic analog demod receiver and validate recovered audio / signal quality.
4. Implement frequency translation and channel filtering for a selected channel.
5. Implement a simple digital demod chain and recover a stable data stream.
6. Simulate digital modulation (BPSK/QPSK) and study constellation behavior under noise.
7. Implement basic packet detection / framing using a preamble / correlation approach.
8. Evaluate link performance and robustness via BER/PER and simple interference mitigation.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Compressive Sensing and Sparse Representation (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Compressive Sensing and Sparse Representation	4	3	0	1	Concepts of Signals and Systems / Mathematics - I

Course Objectives:
- To introduce the principles of sparsity and compressive sensing.
- To understand sparse signal models and reconstruction algorithms.
- To analyze theoretical guarantees for sparse recovery.
- To apply compressive sensing techniques to signal and image processing problems.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain sparsity and compressive sensing fundamentals.
2. Apply sparse representation models to signals.
3. Implement reconstruction algorithms for compressive sensing.
4. Solve practical signal processing problems using CS techniques.

Unit – I **(11 hours)**
Introduction to Compressive Sensing: Motivation for compressive sensing, sparsity and redundancy, sparse signal models, Nyquist sampling limitations, incoherence, measurement matrices.

Unit – II **(12 hours)**
Sparse Representation and Dictionaries: Orthonormal bases, overcomplete dictionaries, dictionary learning, matching pursuit and orthogonal matching pursuit algorithms.

Unit – III **(11 hours)**
Signal Recovery Algorithms Basis pursuit, LASSO, greedy algorithms, convex optimization methods, restricted isometry property (RIP), performance guarantees.

Unit – IV **(11 hours)**
Applications of Compressive Sensing: Compressive sensing in image processing, wireless communications, biomedical signals, radar and sensor networks, recent research trends.

Essential Readings
1. Sparse and Redundant Representations by M. Elad (Springer)
2. Introduction to Linear Algebra by G. Strang (Wellesley-Cambridge Press)
3. A Mathematical Introduction to Compressive Sensing by S. Foucart and H. Rauhut (Birkhäuser).

Suggested Readings

1. An Introduction to Compressive Sampling by E. J. Candès and M. B. Wakin (IEEE Signal Processing Magazine)
2. A Wavelet Tour of Signal Processing by S. Mallat (Academic Press)
3. Compressive Sensing by R. Baraniuk (IEEE Signal Processing Magazine)
4. Sparse and Redundant Representations by M. Elad (Springer)

List of Experiments (Hardware / Software using Matlab / Simulink) **(30 hours)**

1. Verification of sparsity in time and transform domains.
2. Implementation of random measurement matrices.
3. Signal reconstruction using Basis Pursuit.
4. Sparse recovery using Matching Pursuit and OMP.
5. Image reconstruction using compressive sensing.
6. Performance comparison of sparse recovery algorithms.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Wireless System Implementation (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless System Implementation	4	3	0	1	Concepts of Digital Communication Systems, Optical and Wireless Communication

Course Objectives:

- To enable students to design and implement a basic wireless communication system.
- To introduce large-scale wireless propagation models and their role in system design.
- To develop understanding of wireless channel modeling as a linear time-varying (LTV) filter.
- To study statistical multipath channel models used in practical wireless environments.
- To teach digital modulation and demodulation techniques used in wireless communication systems.
- To analyze the performance of digital modulation schemes over Rayleigh fading channels.
- To understand and evaluate spatial diversity techniques for improving reliability.
- To compare single-carrier vs multicarrier (OFDM) modulation, highlighting key trade-offs.
- To provide hands-on experience through a MATLAB + NI USRP based project to validate theoretical concepts experimentally.

Course Outcomes:

After completing the course, the students should be able to:

1. Students are able to analyze the behavior of large scale and small scale wireless propagation models.
2. Students are able to simulate the performance (in Matlab, GNU Radio, LabVIEW) of a wireless communication system in various channel conditions.
3. Students are able to analyze the benefits of multicarrier systems over single carrier systems.
4. Students are able to design and implement a basic wireless communication system using software such as Matlab, GNU Radio or LabVIEW

Unit – I **(11 hours)**

Wireless Propagation and Channel Modeling: Introduction to wireless communication systems, Large-scale propagation models: free-space, two-ray ground reflection, log-distance model, Shadowing effects and path loss estimation, Wireless channel as a linear time-varying (LTV) system, Delay spread, Doppler spread, coherence bandwidth, coherence time, Introduction to multipath propagation

Unit – II **(09 hours)**

Statistical Multipath Fading Models: Small-scale fading mechanisms, Statistical channel models: Rayleigh, Rician, and Nakagami fading, Time-selective and frequency-selective fading, Doppler spectrum and mobility effects, Channel characterization and simulation using MATLAB.

Unit – III **(12 hours)**

Digital Modulation and Diversity Techniques: Digital modulation and demodulation schemes: BPSK, QPSK, M-PSK, M-QAM, BER performance analysis in AWGN and Rayleigh fading channels, Impact of fading on digital communication systems, Spatial diversity techniques: Selection combining, Maximal ratio combining (MRC), Equal gain combining, Performance comparison with and without diversity.

Unit – IV **(13 hours)**

Multicarrier Systems and Practical Implementation: Single-carrier vs multicarrier transmission: trade-offs and design challenges, Introduction to multicarrier modulation, Orthogonal Frequency Division Multiplexing (OFDM): principles and advantages, OFDM system model, cyclic prefix, and PAPR issues, Performance comparison of single-carrier and OFDM systems in fading channels

Course project: MATLAB-based simulation of wireless systems or Hardware/software implementation using NI USRP to validate theoretical concepts

Essential Readings

1. Wireless Communications and Applications Above 100 GHz by T. S. Rappaport *et al.* (IEEE Access)
2. Wireless Communications by A. Goldsmith (Cambridge University Press)
3. Introduction to Wireless Digital Communication: A Signal Processing Perspective by Robert W. Heath Jr. and Jeffrey G. Andrews (Pearson)

Suggested Readings

1. Wireless Communication Systems: From RF Subsystems to 4G Enabling Technologies by Ke – Lin Du, and M. N. S. Swamy (Cambridge University Press)
2. Principles of Mobile Communication by Gordon L. Stuber (Springer)
3. Digital Communications by John G. Proakis and Masoud Salehi (McGraw-Hill)

List of Experiments (Hardware / Software using Matlab / Simulink – at least 5) **(30 hours)**

1. Study and simulation of large-scale path loss and shadowing models in wireless channels.
2. Modeling of wireless channel as a linear time-varying system and analysis of delay and Doppler spread.
3. Simulation of Rayleigh and Rician fading channels and statistical characterization.
4. BER performance analysis of BPSK and QPSK modulation over AWGN and Rayleigh fading channels.
5. Performance evaluation of M-PSK and M-QAM schemes in fading environments.
6. Study and implementation of spatial diversity techniques (Selection Combining and MRC).
7. Performance comparison of single-antenna and diversity-based wireless systems.
8. Simulation and analysis of OFDM systems including cyclic prefix and PAPR effects.
9. Performance comparison of single-carrier and OFDM systems over multipath fading channels.
10. MATLAB/NI-USRP–based implementation and validation of a basic wireless communication link.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Smart Antennas, Beamforming and OTA Testing for 5G/6G (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Smart Antennas, Beamforming and OTA Testing for 5G/6G	4	3	0	1	Concepts of Electromagnetic Theory

Course Objectives:

- To introduce the fundamentals of smart antennas and beamforming techniques used in 5G and 6G systems.
- To develop understanding of antenna arrays and adaptive beamforming algorithms.
- To study massive MIMO and advanced antenna technologies including metasurfaces.
- To familiarize students with OTA testing methodologies and industry standards.
- To provide practical exposure through simulation and measurement-based experiments.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain smart antenna concepts and wireless propagation characteristics for 5G/6G.
2. Analyze and design antenna arrays and beamforming architectures.
3. Apply adaptive and hybrid beamforming algorithms to wireless scenarios.
4. Understand massive MIMO, metasurface and RIS-based antenna systems.
5. Perform OTA testing and interpret standard performance metrics.
6. Use modern RF simulation and measurement tools effectively.

Unit – I **(10 hours)**

Introduction to smart antenna systems and their evolution, limitations of conventional antennas, adaptive and intelligent antenna concepts, advantages in spectral efficiency, coverage and interference suppression, beam steering and beam shaping, spatial diversity and spatial multiplexing, wireless propagation characteristics for sub-6 GHz, mmWave and sub-THz bands.

Unit – II **(12 hours)**

Principles of beamforming and spatial filtering, analog, digital and hybrid beamforming architectures, phase shifter-based and true time delay beamforming, adaptive beamforming concepts, classical algorithms including delay-and-sum, LMS, RLS and MVDR, beam training and beam tracking in 5G NR, codebook-based beamforming.

Unit – III **(11 hours)**

3D beamforming, antenna design challenges for sub-6 GHz and mmWave frequencies, compact wideband antenna arrays, polarization diversity and circular polarization, metasurface-based antennas and reconfigurable intelligent surfaces (RIS), integration issues in base stations and user equipment for 5G and future 6G systems.

Unit – IV **(12 hours)**

Need for OTA testing in modern wireless systems, conducted versus OTA measurements, OTA test environments including anechoic and reverberation chambers, near-field and far-field measurement techniques, compact antenna test ranges, mmWave OTA challenges and calibration of phased arrays, 3GPP OTA standards for 5G NR, emerging OTA requirements for 6G and intelligent radio environments.

Essential Readings

1. Smart Antennas by L. C. Godara (CRC Press)
2. Antenna Theory: Analysis and Design by C. A. Balanis (Wiley)
3. Millimeter Wave Wireless Communications by T. S. Rappaport *et al.* (Pearson)
4. 3GPP TS 38 Series: 5G New Radio Specifications by 3GPP (3GPP)

Suggested Readings

1. Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency by Emil Björnson and Jakob Hoydis (Now Publishers)
2. Fundamentals of Wireless Communication by David Tse and Pramod Viswanath (Cambridge University Press)
3. Phased Array Antenna Handbook by Robert J. Mailloux (Artech House)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Simulation of radiation pattern and array factor of linear antenna arrays.
2. Beam steering and beamwidth analysis of planar antenna arrays.
3. Design and simulation of phased array antennas for sub-6 GHz 5G applications.
4. Analog beamforming using phase shifter-based networks.
5. Digital beamforming implementation using MATLAB.
6. Adaptive beamforming using LMS algorithm.
7. Performance evaluation of MVDR beamforming algorithm.
8. Hybrid beamforming simulation for mmWave 5G systems.
9. Massive MIMO channel modeling and capacity analysis.
10. Design and analysis of compact mmWave antenna arrays.
11. Polarization diversity and circularly polarized antenna array analysis.
12. Metasurface or RIS-based beam control and reconfiguration simulation.
13. Near-field and far-field antenna measurement techniques.
14. OTA measurement of EIRP, TRP and beam patterns.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Enabling Technologies for 6G Communication (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Enabling Technologies for 6G Communication	4	3	0	1	Concepts of Digital Communication Systems, Optical and Wireless Communication

Course Objectives:

- Introduce the vision, requirements, and key performance indicators (KPIs) of 6G wireless communication systems.
- Provide in-depth knowledge of enabling technologies such as THz communication, massive MIMO, AI/ML, RIS, and ultra-low latency networks.
- Explain the role of non-terrestrial networks (NTNs) in future 6G architectures.
- Enable students to analyze CubeSat communication systems, link budgets, and their integration with terrestrial and aerial networks.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain the vision, architecture, and performance requirements of 6G communication systems.
2. Analyze key 6G enabling technologies such as THz bands, massive MIMO, AI-driven networking, and reconfigurable intelligent surfaces.
3. Describe CubeSat systems and communication architectures, including payloads, orbits, and inter-satellite links.
4. Perform basic link budget and performance analysis for CubeSat-based communication systems.
5. Evaluate the integration of CubeSats with 6G terrestrial, aerial (UAV), and non-terrestrial networks for future applications.

Unit – I **(11 hours)**

Introduction to 6G Communication Systems: Evolution from 1G to 6G, Limitations of 5G and motivation for 6G, Vision and KPIs of 6G (Tbps data rates, µs latency, extreme reliability, sustainability), 6G use cases: holographic communication, tactile internet, digital twins, smart cities, 6G network architecture overview, Space–Air–Ground–Sea Integrated Networks (SAGSIN).

Unit – II **(11 hours)**

Enabling Technologies for 6G: Terahertz (THz) communication: spectrum, challenges, channel modelling, Massive MIMO and cell-free MIMO, Reconfigurable Intelligent Surfaces (RIS), AI/ML for 6G: intelligent resource allocation, self-organizing networks, Ultra-low latency and ultra-reliable communication, Energy-efficient and green 6G technologies.

Unit – III **(12 hours)**

CubeSat Systems and Communication Fundamentals: Introduction to CubeSats and small satellites, CubeSat standards and form factors (1U, 3U, 6U, 12U), CubeSat subsystems: structure, power, ADCS, payload, communication, Orbits and coverage (LEO, constellation concepts), CubeSat communication links: Uplink, downlink, inter-satellite links (ISL), Frequency bands (VHF/UHF, S-band, X-band, Ka-band), CubeSat link budget fundamentals

Unit – IV **(11 hours)**

CubeSats in 6G and Future Research Directions: Role of CubeSats in 6G Non-Terrestrial Networks (NTN), Integration of CubeSats with terrestrial, UAV, and HAPS networks, CubeSat-enabled IoT, remote sensing, disaster management, and global connectivity, Challenges: mobility, Doppler, latency, spectrum sharing, security, 6G-enabled CubeSat communication use cases, Open research problems and future trends in 6G-NTN systems

Essential Readings

1. 6G Enabling Technologies: New Dimensions to Wireless Communication by Ramjee Prasad *et al.* (River Publishers / CRC Press)
2. 6G-Enabled Technologies for Next Generation by Amit Kumar Tyagi *et al.* (Wiley)
3. The Road towards 6G: Opportunities, Challenges, and Applications by Valeria Loscri *et al.* (Springer)

Suggested Readings

1. A Vision of 6G Wireless Systems by W. Saad, M. Bennis, and M. Chen (IEEE)
2. What Will 6G Be? by J. G. Andrews *et al*. (IEEE Journal on Selected Areas in Communications)
3. CubeSat Handbook by R. Twiggs and J. Puig-Suari (Academic Press)

<u>List of Experiments</u> (MATLAB / Simulink, STK (optional), Python, NS-3 – at least 5) **(30 hours)**

1. Study of 6G Vision and KPIs
   Analyze key performance indicators of 6G (data rate, latency, reliability, energy efficiency) and compare them with 5G.
2. Path Loss and Channel Modeling for mmWave and THz Bands
   Simulate free-space and atmospheric attenuation models for mmWave/THz frequencies used in 6G.
3. Massive MIMO Performance Analysis
   Evaluate beamforming gain, capacity, and spectral efficiency in massive MIMO systems.
4. Reconfigurable Intelligent Surface (RIS) Assisted Communication
   Simulate RIS-assisted links and analyze SNR and capacity improvement.
5. AI/ML-Based Resource Allocation for 6G Networks
   Implement a simple ML algorithm for power or bandwidth allocation in a 6G-like network scenario.
6. CubeSat Orbit and Coverage Analysis
   Study LEO CubeSat orbits, ground coverage, and revisit time using analytical models or simulation tools.
7. CubeSat Link Budget Calculation
   Compute uplink and downlink link budgets for CubeSat communication in different frequency bands.
8. Doppler Shift and Mobility Effects in CubeSat Communication
   Analyze Doppler effects due to CubeSat mobility and their impact on signal quality.
9. Integration of CubeSat with Terrestrial 6G Network (NTN Scenario)
   Model a basic non-terrestrial network scenario integrating CubeSat and ground users.
10. Mini Project: 6G-Enabled CubeSat Communication System
    Design and simulate a small 6G-NTN use case (IoT via CubeSat, disaster connectivity, or remote sensing).

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Estimation for Wireless Communication (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Estimation for Wireless Communication	4	3	0	1	Concepts of Signals and Systems, Digital Communication Systems, Optical and Wireless Communication

Course Objectives:

- Introduce statistical estimation theory and its importance in wireless communication systems.
- Enable students to understand and apply parameter estimation techniques used in digital and wireless receivers.
- Provide knowledge of channel estimation, synchronization, and detection in fading and noisy wireless channels.
- Expose learners to modern estimation methods used in MIMO, OFDM, and emerging wireless systems.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain the fundamentals of random processes, noise, and estimation theory used in wireless communications.
2. Apply classical estimation techniques such as LS, ML, and MMSE for parameter estimation.
3. Analyze channel estimation and synchronization techniques in wireless systems.
4. Design and evaluate estimation algorithms for modern wireless systems such as OFDM and MIMO.

Unit – I **(12 hours)**

Fundamentals of Estimation Theory: Review of probability and random variables, Random processes and noise models, Parameter estimation problem formulation, Bias, variance, mean square error (MSE), Cramér–Rao Lower Bound (CRLB), Bayesian vs non-Bayesian estimation.

Unit – II **(11 hours)**

Classical Estimation Techniques: Least Squares (LS) estimation, Weighted Least Squares (WLS), Maximum Likelihood (ML) estimation, Minimum Mean Square Error (MMSE) estimation, Linear and nonlinear estimators, Performance comparison of estimators

Unit – III **(11 hours)**

Estimation in Wireless Communication Systems: Channel estimation in AWGN and fading channels, Pilot-based channel estimation, Frequency and phase offset estimation, Timing synchronization, Estimation in OFDM systems, Doppler estimation in mobile channels.

Unit – IV **(11 hours)**

Advanced Estimation Techniques: Estimation in MIMO systems, Adaptive filtering and LMS/RLS algorithms, Kalman filtering for wireless tracking, Sparse channel estimation, Estimation challenges in 5G/6G systems, Introduction to machine learning-based estimation.

Essential Readings

1. An Introduction to Signal Detection and Estimation by H. Vincent Poor (Springer)
2. Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory by Steven M. Kay (Pearson)
3. Adaptive Filter Theory by Simon Haykin (Pearson)

Suggested Readings

1. Detection, Estimation, and Modulation Theory, Part I by Harry L. Van Trees (Wiley)
2. Statistical Signal Processing: Detection, Estimation, and Time Series Analysis by Louis L. Scharf (Addison-Wesley)
3. Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory by Steven M. Kay (Pearson)

List of Experiments (MATLAB / Simulink / Python – at least 5) **(30 hours)**

1. Path Loss and Channel Modeling for mmWave and THz Band
   Simulation of Random Variables and Noise Models
2. Generate Gaussian noise and analyze statistical properties.
   Performance Analysis of Estimators
3. Evaluate bias, variance, and MSE for simple estimators.
4. Least Squares (LS) Estimation
   Implement LS estimation and study its performance.
5. Maximum Likelihood (ML) Estimation
   Implement ML estimator for unknown signal parameters.
6. MMSE Estimation
   Compare MMSE estimator with LS and ML estimators.
7. Channel Estimation in AWGN Channel
   Estimate channel coefficients using pilot symbols.
8. Channel Estimation in Fading Channel
   Analyze estimation performance in Rayleigh fading.
9. OFDM Channel Estimation
   Implement pilot-based channel estimation in OFDM systems.
10. Adaptive Filtering Using LMS Algorithm
    Implement LMS algorithm and analyze convergence behavior.
11. Kalman Filter for Wireless Parameter Tracking
    Track time-varying channel or frequency offset.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Introduction to Nonlinear Circuits (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Nonlinear Circuits	4	3	0	1	Introduction to Electrical and Electronics Engineering, Concepts of Signals and Systems

Course Objectives:

- To analyse nonlinear behavior in circuits such as saturation, hysteresis, switching, bifurcations, and chaos.
- To analyze first/second-order nonlinear dynamics using state equations and phase portraits.
- To connect nonlinear theory to: oscillators, Schmitt triggers / comparators, PLL.
- To understand oscillators as nonlinear systems and study amplitude stabilization.
- Realize nonlinear elements using op-amps/OTAs/CFOAs and impedance-synthesis blocks (NIC/GIC).
- To connect nonlinear theory to switching, triggering, memory circuits, and oscillators.

Course Outcomes:

After completing the course, the students should be able to:

1. Identify sources of nonlinearity in real circuits (device IV, saturation, switching, feedback limits) and explain their impact on behavior.
2. Build nonlinear circuit models using diode / BJT / MOS large-signal elements.
3. Derive state equations for 1-2 state nonlinear circuits and compute / locate equilibrium points and operating regions.
4. Detect and characterize limit cycles in simulation and relate them to oscillator startup and steady-state amplitude.
5. Apply describing-function intuition to estimate oscillation possibility/amplitude.
6. Produce bifurcation style plots using parameter sweeps and recognize period doubling /route-to-chaos signatures in waveforms and spectra.

Unit – I **(10 hours)**

Basics: sources of nonlinearity (device IV, saturation, switching), piecewise-linear / behavioral models, operating regions, linearization. Energy-storage elements (C, L), memristor concept. Realization basics: op-amp/OTA/CFOA as nonlinear blocks and NIC/GIC idea for impedance / nonlinear element emulation.

Unit – II **(12 hours)**

State-space and phase-plane tools: state equations for simple nonlinear RC/RLC circuits, nullclines, equilibrium classification, Lyapunov-style, limit cycles and oscillations, function for predicting oscillation / amplitude. Realization: nonlinear function block (PWL/saturation).

Unit – III **(11 hours)**

Nonlinear oscillators: Van der Pol / Duffing, amplitude stabilization, waveform shaping, Schmitt trigger relaxation oscillators, PLL nonlinearity (pull-in / lock range idea), effect of saturation / clipping on spectra and distortion.

Unit – IV **(12 hours)**

Bifurcations and chaos: period-doubling, routes to chaos, Chua-type circuit and double: scroll attractor, jerk circuit using integrators, memristor - emulator based chaos (NIC / GIC), Poincaré section, parameter sweeps and bifurcation diagrams.

Essential Readings

1. Linear and Nonlinear Circuits by Leon O. Chua, Charles A. Desoer, Ernest S. Kuh (McGraw-Hill)
2. Nonlinear Dynamics and Chaos by Steven H. Strogatz (Westview Press)
3. Chaos in Circuits and Systems by Guanrong Chen and Tetsushi Ueta (World Scientific)
4. Analysis and Design of Analog Integrated Circuits by Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, Robert G. Meyer (Wiley)

Suggested Readings

1. RF Microelectronics by Behzad Razavi (Pearson)
2. Nonlinear Systems by Hassan K. Khalil (Pearson)
3. Synchronization: A Universal Concept in Nonlinear Sciences by Arkady Pikovsky, Michael Rosenblum, Jurgen Kurths (Cambridge University Press)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. To simulate a diode clipper / limiter in time-domain to observe waveform clipping and quantify harmonic distortion using FFT.
2. To implement a Van der Pol type oscillator to verify limit-cycle formation and amplitude stabilization from different initial conditions.
3. To simulate a Duffing oscillator under periodic forcing to demonstrate hysteresis/jump phenomenon by sweeping the drive frequency up and down.
4. To apply describing-function style prediction to a nonlinear amplifier and linear filter loop to estimate oscillation condition/amplitude and validate in simulation.
5. To build a Chua-like nonlinear oscillator model to capture chaotic signatures using phase portraits and time series plots.
6. To run a parameter sweep on a nonlinear oscillator coefficient (gain/nonlinearity) to generate a bifurcation-style plot and identify period changes.
7. To perform Poincaré sampling of a periodically forced nonlinear system to construct a Poincaré map and interpret periodic vs chaotic behavior.
8. To build a memristor emulator using NIC/GIC concepts (behavioral or op-amp implementation) to verify pinched hysteresis and explore memristive chaos.
9. To implement a jerk circuit to generate chaotic attractors.
10. To implement an OTA-based voltage-controlled resistor (VCR) / nonlinear transconductor to demonstrate controllable damping/negative resistance in a dynamic circuit.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

RF Integrated Circuit Design (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
RF Integrated Circuit Design	4	3	0	1	Concepts of Electromagnetic Theory / Microwave Integrated Circuits

Course Objectives:

- To introduce a practical RFIC design flow.
- To design core RFIC receiver blocks with noise / linearity studies.
- To design LC VCOs and PLL-based frequency synthesizers and relate phase noise/jitter to system performance.
- To learn RFIC layout practices and post-layout correlation.

Course Outcomes:
After completing the course, the students should be able to:

1. Translate RF transceiver requirements into block-level targets using gain / noise / linearity budgets.
2. Design and verify mixers / VGAs / baseband filters and evaluate conversion gain, NF, IIP3, and LO leakage trends.
3. Design an LC VCO and PLL synthesizer, and evaluate phase noise/jitter impact on receiver/transmitter performance.
4. Apply RFIC layout techniques for passives and sensitive nodes.

Unit – I **(10 hours)**
RFIC System: RFIC vs MIC (substrate loss, parasitics, Q, coupling), compact RF models, passives (spiral L, MIM C, varactors, resistors), floorplanning and isolation (guard rings, shielding, differential routing).

Unit – II **(12 hours)**
Receiver IC Blocks: direct-conversion/low-IF comparison, mixers (passive/active), Metrics: conversion gain / loss, NF, IIP3, isolation, LO feedthrough), VGAs and AGC concept, baseband filtering (active-RC / Gm-C), DC offset and flicker-noise.

Unit – III **(12 hours)**
Frequency Synthesis and LO Generation: LC VCO design, phase noise, LO buffers, quadrature LO generation, PLL blocks and lock behavior.

Unit – IV **(11 hours)**
RFIC Integration: Layout (passive placement, routing parasitics, substrate coupling, supply/ground noise), EM simulation, corner / Monte-Carlo robustness checks, stability / oscillation checks for RF blocks.

Essential Readings

1. RF Microelectronics by Behzad Razavi (Pearson)
2. The Design of CMOS Radio-Frequency Integrated Circuits by Thomas H. Lee (Cambridge University Press)
3. Phase-Locked Loops: Design, Simulation, and Applications by Roland E. Best (McGraw-Hill)
4. CMOS Integrated Circuit Design by Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer (Wiley)

Suggested Readings

1. Microwave Transistor Amplifiers: Analysis and Design by Guillermo Gonzalez (Pearson)
2. RF Circuit Design: Theory and Applications by Reinhold Ludwig and Pavel Bretchko (Pearson)
3. RF Power Amplifiers for Wireless Communications by Steve C. Cripps (Artech House)

List of Experiments (Hardware / Software) **(30 hours)**

1. To model on chip passives and interconnect parasitic to evaluate their impact on resonance, Q, and matching in an RF block.
2. To design a frequency-translation block to analyze key metrics.
3. To design a variable-gain stage to verify gain range and observe noise/linearity trade-offs.
4. To design an oscillator block to verify start-up, tuning range, amplitude, and phase-noise trends.
5. To implement a frequency synthesizer loop to evaluate lock behavior.
6. To design an LO distribution / buffering block to validate isolation / loading effects and their influence on oscillator/synthesizer performance.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Semiconductor Device Characterization (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Semiconductor Device Characterization	4	2	0	2	NIL

Course Objectives:

- Perform DC, pulsed, and RF characterization of semiconductor devices (diodes, MOSFETs, HEMTs, BJTs).
- Extract key parameters (V_{th}, g_m, r_o, R_{ON}, capacitances, f_T / f_{max}) and build compact models for circuit simulation.
- Understand and execute reliability / ageing tests (BTI, HCI, TDDB, EM, self-heating, trapping) and interpret degradation trends.
- Learn measurement automation + data analysis using Python / Matlab and correlation with SPICE / TCAD / compact models.

Course Outcomes:
After completing the course, the students should be able to:

1. Set up semiconductor device measurements using SMU / LCR, Kelvin sensing, calibration.
2. Perform DC and CV characterization of diodes / BJTs / MOSFETs / HEMTs and extract key parameters such as V_{TH}, g_M, r_o, R_{ON}, contact resistance, and capacitances.
3. Carry out pulsed and transient measurements to separate trapping and self-heating effects and extract dynamic R_{ON}, time constants, and thermal RC.
4. Perform RF S-parameter measurements with basic calibration and de-embedding and extract small-signal models and f_T / f_{max} proxies for device evaluation.
5. Execute accelerated reliability stress tests (BTI, HCI, TDDB awareness, self-heating) and interpret ageing / degradation trends with simple lifetime extrapolation.

Unit – I **(15 hours)**
Measurement: Test setups (SMU / curve tracer / LCR), 2-wire vs 4-wire (Kelvin) sensing, guarding and leakage control, ESD precautions, probe station basics, contact resistance and chuck effects, calibrations.

DC / CV characterization: Diode / FET IV curves, threshold extraction, g_M/r_o extraction, subthreshold slope and mobility, R_{ON} and contact resistance, CV measurements, compact model for SPICE.

Unit – II **(15 hours)**
Pulsed / Transient Characterization: pulsed IV and self-heating, trap characterization using stress, recovery transients, dynamic R_{ON} extraction, gate lag / drain lag, charge trapping and time constants (stretched-exponential fitting).

RF Characterization: S-parameters interpretation, calibration basics (SOLT / TRL), de-embedding, small-signal equivalent, f_T / f_{max}. Reliability: accelerated ageing and lifetime extrapolation (Arrhenius), Weibull Plots.

Essential Readings

1. Semiconductor Device Fundamentals by Robert F. Pierret (Addison-Wesley)
2. Semiconductor Devices: Physics and Technology by S. M. Sze and Ming-Kwei Lee (Wiley)
3. CMOS Circuit Design, Layout, and Simulation by R. Jacob Baker (Wiley-IEEE Press)
4. RF Microelectronics by Behzad Razavi (Pearson)

Suggested Readings

1. Microwave Transistor Amplifiers: Analysis and Design by Guillermo Gonzalez (Pearson)
2. Integrated Circuit Failure Analysis: A Guide to Preparation Techniques by Friedrich Beck and Stephen S. Wilson (Wiley)
3. Analysis and Design of Analog Integrated Circuits by Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer (Wiley)

List of Experiments (Hardware / Software – at least 5) **(60 hours)**

1. To measure diode IV characteristics to extract R_s, Ideality factor (η), and leakage current (and temperature dependence).
2. To measure MOSFET DC characteristics (I_D-V_G and I_D-V_D) to extract V_{TH}, g_M, r_o, and subthreshold slope.
3. To measure R_{ON} versus V_G to estimate ON - Resistance and contact resistance.
4. To perform CV measurements to extract bias dependent capacitances (C_{GS} / C_{GD} / C_{DS}).
5. To set up pulsed IV measurements with chosen pulse width and duty cycle.
6. To compare DC and pulsed measurements to extract dynamic R_{ON} and trapping time constants.
7. To apply gate / drain stress and then measure recovery transients to quantify gate lag / drain lag behavior and trapping signatures.
8. To measure bias-dependent S-parameters of a transistor at multiple bias points.
9. To extract small-signal model parameters (C_{GS}, C_{GD}, R_G, R_S, R_D) and estimate f_T and validate using SSEC Model.
10. To execute stress–measure–recover sequences to extract V_{TH} shift versus time.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Compact Modeling with Verilog-A (DSE – 10 / GE - 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Compact Modeling with Verilog-A	4	2	0	2	Introduction to Electrical and Electronics Engineering, Concepts of CMOS VLSI Design

Course Objectives:
- Develop practical compact models for R / C / L.
- Build compact models for MOSFET and HEMT.
- Implement models in Verilog-A with convergence checks.
- Package models into a mini-PDK with parameter forms and layout generators for devices.

Course Outcomes:
After completing the course, the students should be able to:
1. Develop compact models for resistors, capacitors and inductors and validate them across DC and frequency ranges.
2. Build MOSFET compact models capturing modern architecture trends and verify continuity and charge behavior in simulations.
3. Implement progressive HEMT compact models and validate DC, and small signal trends.
4. Deliver a mini-PDK: Verilog-A models with parameterized layout cells that reflect specifications.

Unit – I **(15 hours)**
Resistor Models: sheet resistance, geometry scaling, contact resistance, temperature coefficient. Capacitor models, frequency. Inductor models, substrate loss, Q-factor, self-resonance, RLC equivalents. Parameter extraction.

Compact Modeling for MOSFET: IV equations, CLM, mobility degradation / velocity saturation, body effect, charge / capacitance partitioning. Parameter extraction.

Unit – II **(15 hours)**
HEMT Modeling: knee behavior, access resistances, nonlinear capacitances, transconductance, breakdown, dispersion. Angelov large signal /ASM / MVSG surface potential / charge based model.

Verilog-A compact models, DC / AC / transient, parameter sweeps, symbols, parameter forms. Layout generation: parameterized layouts for resistor arrays, capacitor shapes, spiral inductors, multi-finger devices.

Essential Readings

1. Operation and Modeling of the MOS Transistor by Yannis Tsividis & Colin McAndrew (Oxford University Press)
2. Device Modeling for Analog and RF CMOS Circuit Design by Trond Ytterdal, Yngvar Berg & Tor A. Fjeldly (Wiley)
3. GaN Transistors for Efficient Power Conversion by Alex Lidow, Johan Strydom, Michael de Rooij & David Reusch (Wiley)
4. Verilog-A Language Reference Manual by Accellera Systems Initiative (Accellera)

Suggested Readings

1. The SPICE Book by Andrei Vladimirescu (Wiley)
2. RF Microelectronics by Behzad Razavi (Pearson)
3. ADS Documentation (PDK / AEL / Model Integration) by Keysight Technologies (Keysight)

List of Experiments (Hardware / Software – at least 5) **(60 hours)**

1. To extract a resistor model to validate geometry scaling and temperature behavior.
2. To build a capacitor compact model and observe CV / impedance-vs-frequency trends.
3. To build an inductor compact model and observe trends.
4. To implement the passive models in Verilog-A to and run DC / AC / transient simulations.
5. To extract a MOSFET compact model parameter set from I_D-V_G / I_D-V_D / g_M / r_o curves.
6. To extend the MOSFET model with simple capacitance model for switching transients.
7. To implement a baseline HEMT compact model to fit DC output / transfer characteristics.
8. To implement a baseline HEMT compact model to fit small-signal trends.
9. To add trapping / dispersion and self-heating to the HEMT model.
10. To create parameterized layout generators for active / passive devices.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

IoT Sensor and Data Integration (DSE – 9 / GE - 9)
(Credit Distribution and Prerequisites of the Course)

<table>
<tr><th rowspan="2">Course Title</th><th rowspan="2">Credits</th><th colspan="3">Credit Distribution of the Course</th><th rowspan="2">Prerequisite of the course (if any)</th></tr>
<tr><th>Lecture</th><th>Tutorial</th><th>Practical</th></tr>
<tr><td>IoT Sensor and Data Integration</td><td>4</td><td>3</td><td>0</td><td>1</td><td>Introduction to Embedded Systems for IoT</td></tr>
</table>

Course Objectives:

- To understand practical sensor characteristics and select sensors using datasheets for IoT applications.
- To interface analog and digital sensors and build acquisition modules for embedded systems.
- To apply basic signal conditioning and filtering to improve measurement quality.
- To perform calibration, error estimation, and validation for reliable sensor data.

Course Outcomes:

After completing the course, the students should be able to:

1. Select sensors and interfaces for IoT nodes by interpreting range, accuracy, resolution, and bandwidth from datasheets.
2. Interface analog/digital sensors and implement a stable sampling pipeline with timing and timestamping.
3. Apply basic conditioning and filtering techniques to reduce noise and improve data quality.
4. Perform calibration (offset/gain) and prepare simple error/uncertainty notes for measurements.
5. Build a multi-sensor data integration prototype producing clean, validated, and well-structured datasets.

Unit – I **(10 hours)**

Sensor fundamentals for IoT: sensor types, static and dynamic characteristics, response time and bandwidth, analog vs digital sensors, interface choices (analog / ADC, I2C, SPI, UART).

Unit – II **(11 hours)**

Signal Conditioning: scaling and offset, voltage dividers / bridges concept, buffering, RC filtering and anti-aliasing, ADC (reference, sampling time, quantization), sampling strategies (polling vs interrupt vs timer-triggered).

Unit – III **(12 hours)**

Noise and filtering: noise sources, smoothing, moving average / median, simple IIR filter, sampling rate, timestamping and jitter, unit conversion and scaling, data validation (range checks, missing samples), sensor fusion.

Unit – IV **(12 hours)**

Calibration and Validation: one-point / two-point calibration, offset / gain correction, temperature drift observation, repeatability and simple uncertainty notes, storing calibration coefficients, multi-sensor synchronization and data integration.

Essential Readings

1. Sensors and Actuators: Engineering System Instrumentation by Clarence W. de Silva (CRC Press)
2. Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti (Orient Blackswan Private Limited)
3. Practical Electronics for Inventors by Paul Scherz and Simon Monk (McGraw-Hill Education)
4. Measurement Systems: Application and Design by Ernest O. Doebelin and Dhanesh N. Manik (McGraw-Hill Education)

Suggested Readings

1. Building Wireless Sensor Networks by Robert Faludi (O'Reilly Media)
2. Data Acquisition Handbook by Omega Engineering (Omega Press)
3. Embedded Systems: Real-Time Interfacing to ARM Cortex-M Microcontrollers by Jonathan W. Valvano (Amazon)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. To interface a digital sensor using I2C / SPI / UART and acquire periodic measurement.
2. To interface an analog sensor using ADC and implement fixed-rate sampling using timer-based acquisition.
3. To implement basic hardware signal conditioning using an RC low-pass filter and compare noise levels before and after conditioning.
4. To implement digital filtering methods (moving average and median) and compare smoothing vs response time on sensor data.
5. To measure sampling jitter and timing stability for polling-based sampling versus timer/interrupt-based sampling.
6. To perform two-point calibration for a sensor (offset and gain correction) and store calibration coefficients in non-volatile memory.
7. To study drift and stability by long-duration logging and quantify drift trends and repeatability.
8. To implement a basic sensor fusion using an IMU to estimate tilt / orientation using a complementary filter.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

IoT Reliability Engineering (DSE – 9 / GE - 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
IoT Reliability Engineering	4	3	0	1	Introduction to IoT, Basics of Networks

Course Objectives:

- To introduce reliability engineering concepts in the context of IoT systems.
- To understand failure mechanisms and reliability metrics for IoT devices and networks.
- To study fault-tolerant and resilient IoT architectures.
- To familiarize students with reliability testing, validation and lifecycle management.
- To address safety and security aspects affecting IoT system reliability.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain reliability, availability and maintainability concepts for IoT systems.
2. Analyze failure modes and reliability metrics of IoT architectures.
3. Design fault-tolerant and resilient IoT systems.
4. Apply reliability testing and validation techniques to IoT devices.
5. Evaluate the impact of security and safety on IoT reliability.
6. Develop reliability-aware strategies for large-scale IoT deployments.

Unit – I **(12 hours)**

Concept of reliability, availability, maintainability and safety (RAMS), reliability metrics such as MTBF, MTTR, failure rate and hazard function, bathtub curve, differences between conventional electronic systems and IoT systems, reliability block diagrams applied to IoT architectures.

Unit – II **(12 hours)**

Fault classification, transient, intermittent and permanent faults, redundancy techniques including hardware, software and information redundancy, fault detection, isolation and recovery (FDIR), watchdog timers and checkpointing, graceful degradation and self-healing IoT systems, resilient network topologies.

Unit – III **(11 hours)**

Design for reliability (DfR) and design for testability (DfT) in IoT devices, accelerated life testing, highly accelerated life testing (HALT) and highly accelerated stress screening (HASS), failure analysis techniques including FMEA and FTA, reliability prediction standards.

Unit – IV **(10 hours)**

Interdependence of security and reliability in IoT systems, impact of cyber-attacks on system dependability, secure boot and trusted execution environments, redundancy for secure communications, safety-critical IoT systems, standards and certifications for industrial IoT reliability, case studies from smart grid, healthcare IoT and autonomous systems.

Essential Readings

1. Reliability Engineering by Elsayed A. Elsayed (Wiley)
2. An Introduction to Reliability and Maintainability Engineering by Charles E. Ebeling (McGraw-Hill)
3. Engineering a Safer World by Nancy Leveson (MIT Press)
4. Practical Reliability Engineering by Patrick D. T. O'Connor and Andre Kleyner (Wiley)

Suggested Readings

1. Reliability Engineering and Risk Analysis: A Practical Guide by Mohammad Modarres *et al.* (CRC Press)
2. Maintenance Theory of Reliability by Toshio Nakagawa (Springer)
3. Internet of Things – From Research and Innovation to Market Deployment by Ovidiu Vermesan and Peter Friess (River Publishers)
4. Empowering IoT: Reliability, Network Management, Sensing, and Probabilistic Charging in Wireless Sensor Networks by Muhammad Umar Farooq Qaisar , Weijie Yuan , Paolo Bellavista , Hina Tabassum (Springer)

List of Experiments (Hardware / Software) **(30 hours)**

1. Fault injection and observation in embedded IoT firmware.
2. Implementation of watchdog timers for fault detection.
3. Redundancy implementation for fault tolerance in IoT nodes.
4. Reliability-aware routing simulation in IoT networks.
5. Stress testing of IoT hardware under thermal and voltage variations.
6. Accelerated life testing simulation for IoT components.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

LPWAN IoT Network (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
LPWAN IoT Network	4	3	0	1	Introduction to IoT

Course Objectives:

- To understand LPWAN vs short-range wireless vs cellular IoT based on range, power, and latency.
- To learn LoRa / LoRaWAN through configuration, join, and end-to-end data transfer labs.
- To introduce mesh networking for IoT with multi-hop validation.
- To perform simple link and field testing (RSSI / SNR, range, airtime, payload planning).
- To integrate nodes with a gateway / server → MQTT → application flow.

Course Outcomes:
After completing the course, the students should be able to:

1. Compare IoT wireless options and justify a connectivity choice for a given industrial use-case.
2. Configure LoRa links and operate a LoRaWAN node using a network server.
3. Build a small mesh network and validate multi-hop delivery, latency, and power trade-offs.
4. Plan payload size, duty cycle, and transmission strategy using simple experiments and link metrics.
5. Demonstrate an end-to-end LPWAN / mesh system using a gateway/server bridged to MQTT.

Unit – I **(10 hours)**
Connectivity Requirements: range, data rate, latency, reliability, power, cost; star vs mesh; Wi-Fi / BLE vs LPWAN and cellular IoT (NB-IoT / LTE-M); device identity and provisioning; link metrics (RSSI / SNR).

Unit – II **(12 hours)**
LoRa PhyLayer (chirp spread spectrum), LoRa vs LoRaWAN stack, OTAA / ABP, device classes A / B / C, confirmed / unconfirmed messages, payload sizing, airtime awareness, ADR, gateway and network server.

Unit – III **(12 hours)**
Mesh Fundamentals: roles, commissioning, addressing / routing intuition, latency vs reliability trade-offs, sleep-friendly operation, gateway / border router concept, integrating sensing and command / control.

Unit – IV **(11 hours)**
Gateway /server to MQTT bridge, payload formats, range testing (RSSI / SNR), interference considerations, power budgeting for long-life nodes, multi-node LPWAN / mesh system with end-to-end data delivery.

Essential Readings

1. LoRaWAN 1.0.4 Specification by LoRa Alliance (LoRa Alliance)
2. LoRaWAN System Design Guidelines by LoRa Alliance (LoRa Alliance)
3. Building Wireless Sensor Networks: A Practical Guide to the Zigbee Mesh Networking Protocol by Robert Faludi (O'Reilly Media)
4. MQTT Essentials by HiveMQ (HiveMQ)

Suggested Readings

1. Zigbee Wireless Networking by Drew Gislason (Newnes/Elsevier)
2. Bluetooth Low Energy: The Developer's Handbook by Robin Heydon (Prentice Hall)
3. Designing Connected Products by Claire Rowland *et al.* (O'Reilly Media)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. To establish a LoRa point-to-point link and log RSSI/SNR across different distances and environments.
2. To configure a LoRaWAN node and perform OTAA join and uplink using a network server console
3. To study payload and airtime effects by varying payload size and transmit interval.
4. To conduct range test and study coverage and reliability using packet success rate and link metrics.
5. To bridge LoRaWAN uplinks to an MQTT and verify message format.
6. To form a small mesh network and validate multi-hop delivery.
7. To compare sleep vs always-on operation in a mesh / LPWAN node.
8. To implement a downlink / command-control action (e.g., toggle LED/relay).

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Edge Computing (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Edge Computing	4	3	0	1	Fundamentals of Computer Programming

Course Objectives:

- Introduce the fundamental concepts, architecture, and motivation behind edge computing and its evolution from cloud and fog computing.
- Enable students to understand edge computing architectures, devices, and communication technologies used in real-time and distributed systems.
- Provide knowledge of resource management, performance optimization, and security challenges in edge computing environments.
- Expose learners to practical applications, industry use cases, and emerging trends such as Edge-AI, IoT integration, and 5G/6G-enabled edge systems.

Course Outcomes:

After completing the course, the students should be able to:

1. Explain the concepts, characteristics, and advantages of edge computing and compare it with cloud and fog computing.
2. Describe and analyze edge computing architectures, components, and data flow for various real-time applications.
3. Apply techniques for resource allocation, latency reduction, performance optimization, and security in edge environments.
4. Identify and evaluate edge computing applications in domains such as IoT, smart cities, healthcare, vehicular networks, and Industry 4.0.
5. Analyze current research trends and future challenges in edge computing, including Edge-AI and next-generation wireless networks.

Unit – I **(11 hours)**

Introduction & Fundamentals: Edge Computing - definition, Evolution from Cloud & Fog Computing, Key features: low latency, distributed processing, real-time decisioning, Types of Edge, Edge vs Cloud vs Fog Computing comparison, Use cases and application domains.

Unit – II **(12 hours)**

Architecture & Core Technologies: Edge computing architecture and models, Edge devices, gateways, and micro data centers, Communication protocols & connectivity, Computation offloading and distributed analytics, Multi-Access Edge Computing (MEC) and infrastructure components.

Unit – III **(11 hours)**

Resource Management, Performance & Security: Resource allocation, scheduling, load balancing in edge environments, Measuring performance: latency, throughput, energy efficiency, Data management at the edge, Security challenges, Edge analytics and adaptive systems.

Unit – IV **(11 hours)**
Applications, Current Trends & Future Directions: Industry use cases: smart factories, connected vehicles, real-time AI/ML at edge, Edge AI / Edge intelligence frameworks, Integration with IoT & 5G ecosystems, Emerging paradigms.

Essential Readings

1. Fog and Edge Computing: Principles and Paradigms by Rajkumar Buyya and Satish Narayana Srirama (Wiley)
2. Mobile Edge Computing by Yan Zhang (Springer)
3. Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi (Morgan Kaufmann)

Suggested Readings

1. Edge Computing Fundamentals, Advances and Applications by Anitha Kumari, G. Sudha Sadasivam, D. Dharani & M. Niranjanamurthy (CRC Press)
2. Edge Computing: A Primer by J. Cao, Q. Zhang, and W. Shi (Springer)
3. Edge Computing: From Hype to Reality by Fadi Al-Turjman (Springer)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. Setting Up an Edge Device
   Installation and configuration of an edge device (Raspberry Pi / Jetson Nano / Virtual Edge Node) and required software tools.
2. Data Acquisition at the Edge
   Collect sensor data (temperature, humidity, motion, etc.) using an edge device and store it locally.
3. Edge Data Filtering and Aggregation
   Design and implement data filtering, compression, or aggregation techniques at the edge to reduce network load.
4. Computation Offloading in Edge Computing
   Implement a simple task offloading mechanism between edge device and cloud server and analyze performance.
5. Resource Monitoring and Management at the Edge
   Monitor CPU, memory, and energy consumption of edge devices during application execution.
6. Security Implementation at the Edge
   Implement basic security mechanisms such as authentication, encryption, or secure data transmission for edge applications.
7. Edge Analytics / Edge AI Application
   Deploy a lightweight machine learning model or rule-based analytics on an edge device for real-time decision-making.
8. Mini Project: Real-Time Edge Application
   Design and implement a small edge computing application (smart home, health monitoring, traffic sensing, or industrial monitoring).

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Forensic Image Processing (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Forensic Image Processing	4	3	0	1	Fundamentals of Computer Programming, Fundamentals of Image Processing

Course Objectives:

- Build foundational understanding of images / videos as digital evidence, including forensic acquisition, metadata integrity, hashing.
- Develop practical skills to detect and localize classical image tampering (copy-move, splicing).
- Introduce learning-based forensic methods for manipulation and deepfake detection.
- Provide exposure to forensic biometrics and video forensics.

Course Outcomes:
After completing the course, the students should be able to:

1. Inspect metadata and compression traces to infer source and manipulation likelihood.
2. Detect common tampering (copy-move, splicing) and localize manipulated regions.
3. Evaluate deepfake and synthetic media detectors and interpret failure modes.
4. Apply and evaluate learning-based manipulation/deepfake detection approaches using key metrics.

Unit – I **(12 hours)**
Forensic Foundations: Role of images and videos as digital evidence, forensic image acquisition, image formats: EXIF and metadata integrity, hash-based integrity checks, provenance basics. JPEG compression pipeline, double-compression artifacts, resampling clues, noise patterns.

Unit – II **(11 hours)**
Classical Tamper Detection & Localization: Copy-move detection (block matching / feature matching), splicing cues, illumination inconsistencies, edge artifacts, color filter array (CFA) clues overview, error level analysis (ELA) and when it fails. Producing tamper heatmaps and tuning thresholds responsibly.

Unit – III **(11 hours)**
Learning-Based Forensics: CNN-based manipulation detection, patch-level classifiers, localization networks overview, evaluation metrics (AUC, F1, pixel IoU). Deepfake basics: face swap, reenactment, diffusion-based edits.

Unit – IV **(11 hours)**
Forensic Biometrics: Face, Fingerprint, Iris, Gait; feature extraction and matching; video forensics; frame tampering; camera identification. Watermarking overview (visible/invisible, fragile/robust) and steganalysis, content authenticity initiatives.

Essential Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson)
2. Photo Forensics by H. Farid (MIT Press)
3. Digital Image Forensics: There is More to a Picture Than Meets the Eye by Husrev Taha Sencar and Nasir Memon (Springer)

Suggested Readings

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press)
2. Computer Vision: Algorithms and Applications by Richard Szeliski (Springer)
3. Practical Cryptography by Niels Ferguson and Bruce Schneier (Wiley)

List of Experiments (Hardware / Software) **(30 hours)**

1. To extract and analyze EXIF / metadata and generate hash checksums to document integrity and provenance indicators for a given image set.
2. To vary JPEG quality and recompress images using FFmpeg/OpenCV to study visible and statistical compression artifacts across quality factors.
3. To detect double JPEG compression signatures to flag probable recompression and compare true/false positives on a small dataset.
4. To implement copy-move detection using feature matching (ORB/SIFT) to localize duplicated regions and generate a tamper heatmap.
5. To train a patch-based tamper classifier to evaluate AUC/F1 and analyze failure cases and bias.
6. To perform robustness testing by applying resize/recompress/blur perturbations to quantify detector sensitivity and produce a robustness table.
7. Watermarking and Steganalysis basics.
8. Video frame extraction and tampering detection.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Computer Graphics (DSE – 9 / GE – 9)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Computer Graphics	4	3	0	1	Concepts of Data Structures

Course Objectives:
- Introduce fundamental concepts of computer graphics.
- Explain key algorithms and techniques for creating graphics.
- Cover generation, manipulation, and display of 2D & 3D objects.
- Build mathematical foundations used in graphics systems.
- Teach rendering techniques used in modern graphics.
- Understand and apply graphical transformations (2D/3D).

Course Outcomes:
After completing the course, the students should be able to:
1. Understand the basic principles and architecture of computer graphics systems.
2. Apply algorithms for drawing and transforming 2D and 3D objects.
3. Implement clipping, viewing, and projection techniques.
4. Understand visible surface detection, illumination, and shading models.
5. Apply computer graphics concepts in practical applications and software tools.

Unit – I **(11 hours)**
Introduction and Graphics Systems: Introduction to computer graphics and applications, Graphics hardware and software, Display devices: raster scan and random scan displays, Input devices and interaction techniques, Graphics primitives and scan conversion, Line drawing algorithms: DDA and Bresenham, Circle and ellipse drawing algorithms.

Unit – II **(12 hours)**
Two-Dimensional Graphics and Transformations: 2D coordinate systems and homogeneous coordinates, 2D transformations: translation, rotation, scaling, reflection, shearing, Composite transformations, Windowing and viewing transformations, Line and polygon clipping algorithms: Cohen–Sutherland, Liang–Barsky, Sutherland–Hodgman.

Unit – III **(11 hours)**
Three-Dimensional Graphics and Viewing: 3D coordinate systems and transformations, 3D transformations and composite transformations, Parallel and perspective projections, Viewing pipeline and 3D viewing transformations, Hidden surface removal techniques: Z-buffer, back-face culling, Introduction to curves and surfaces: Bézier curves and B-splines.

Unit – IV **(11 hours)**
Illumination, Shading, and Advanced Topics: Color models: RGB, CMY, HSV, Illumination models: ambient, diffuse, and specular reflection, Shading techniques: flat, Gouraud, and Phong shading, Texture mapping and anti-aliasing, Introduction to animation and rendering pipeline, Overview of modern graphics: OpenGL concepts, ray tracing basics.

Essential Readings

1. Computer Graphics: Principles and Practice by Foley *et al.* (Pearson)
2. Fundamentals of Computer Graphics by Peter Shirley *et al.* (CRC Press)
3. Interactive Computer Graphics: A Top-Down Approach by Shader-Based OpenGL by Edward Angel and Dave Shreiner (Pearson)

Suggested Readings

1. Computer Graphics with OpenGL by D. Hearn & M. P. Baker (Pearson)
2. Procedural Elements for Computer Graphics by D. F. Rogers (McGraw-Hill)
3. Computer Graphics by Zhigang Xiang (McGraw-Hill)

List of Experiments (Hardware / Software) **(30 hours)**

1. Implementation of DDA and Bresenham line drawing algorithms.
2. Implementation of Midpoint circle and ellipse drawing algorithms.
3. Program to perform 2D transformations (translation, rotation, scaling) on graphical objects.
4. Implementation of composite 2D transformations using homogeneous coordinates.
5. Implementation of line clipping algorithms (Cohen–Sutherland and Liang–Barsky).
6. Implementation of polygon clipping using Sutherland–Hodgman algorithm.
7. Program to perform 3D transformations and projections (parallel and perspective).
8. Implementation of hidden surface removal using Z-buffer algorithm.
9. Implementation of illumination and shading models (Phong/Gouraud shading).
10. Design of a simple OpenGL-based graphics application demonstrating rendering and animation.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Computational Photography and Mobile Imaging (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Computational Photography and Mobile Imaging	4	3	0	1	Fundamentals of Image Processing or Digital Image Processing

Course Objectives:

- Understand the digital camera imaging pipeline and implement key Image Signal Processing stages used in mobile cameras.
- Design practical enhancement pipelines for color, tone, noise, blur and dynamic range.
- Build robust multi-image computational photography features: HDR, panorama, super-resolution.
- Evaluate photo quality using simple objective metrics and / or perceptual checks.

Course Outcomes:
After completing the course, the students should be able to:

1. Acquire and represent images using sampling / quantization, and explain the basic camera / ISP pipeline and color fundamentals.
2. Enhance images using spatial and frequency domain methods for contrast, smoothing, and sharpening with artifact control.
3. Apply restoration methods for noise reduction and deblurring, with practical parameter tuning.
4. Build pipelines for registration, HDR/exposure fusion, and panorama stitching with blending.
5. Analyze compression artifacts and assess quality using PSNR/SSIM and perceptual checks.

Unit – I **(10 hours)**
Digital Image Fundamentals: DIP system, image sensing, acquisition, sampling, quantization, basic radiometry / illumination-reflectance. Color models and transformations, white balance, gamma correction, camera overview (RAW vs JPEG, Bayer pattern).

Unit – II **(12 hours)**
Image Enhancement-Spatial Domain: Point processing, intensity transformations, histogram processing, spatial filtering. Edge-preserving enhancement, local contrast enhancement for mobile photos (CLAHE).

Unit – III **(12 hours)**
Image Enhancement-Frequency Domain & Image Restoration: 2D DFT, frequency-domain filtering, homomorphic filtering for illumination correction. Image degradation / noise models, restoration filters, deblurring, practical denoise - sharpen trade-offs for low-light mobile imaging.

Unit – IV **(11 hours)**
Computational Photography: Image registration, panorama stitching, HDR and exposure fusion, super-resolution via multi-frame fusion, compression in mobile imaging (JPEG/HEIF), image quality assessment.

Essential Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson)
2. Computer Vision: Algorithms and Applications by Richard Szeliski (Springer)
3. The Image Processing Handbook by John C. Russ (CRC Press)
4. Handbook of Image and Video Processing by Alan C. Bovik (Academic Press)

Suggested Readings

1. Digital Image Processing by William K. Pratt (Wiley)
2. Learning OpenCV by Gary Bradski and Adrian Kaehler (O'Reilly Media)
3. Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman (Cambridge University Press)

List of Experiments (Hardware / Software – at least 5) **(30 hours)**

1. To implement gamma correction, log transform, and piecewise linear contrast.
2. To implement histogram equalization.
3. To implement spatial smoothing filters.
4. To implement sharpening (Laplacian, unsharp mask, high-boost) on test images.
5. To implement frequency-domain low-pass and high-pass filtering and to compare with spatial filtering.
6. To simulate motion blur / defocus blur and additive noise and to restore images using a Wiener-style deconvolution baseline.
7. To implement feature-based image alignment and to generate aligned stacks from handheld shots.
8. To build panorama stitching.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Image Restoration and Inpainting (DSE – 10 / GE – 10)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Image Restoration and Inpainting	4	3	0	1	Fundamentals of Image Processing or Digital Image Processing

Course Objectives:
- Develop the ability to model image degradations (noise, blur, compression).
- Implement denoising and deblurring workflow with parameter tuning.
- Build inpainting /object removal workflows using classical method.
- Evaluate and report restoration quality using metrics and perceptual checks.

Course Outcomes:
After completing the course, the students should be able to:
1. Create degradation models (noise / blur / compression) and generate synthetic test cases and datasets for fair evaluation.
2. Implement and compare denoising methods and justify parameter choices using metrics and visual artifact inspection.
3. Apply deblurring / deconvolution (non-blind and blind workflows) and control ringing and amplification artifacts.
4. Perform inpainting / object removal using patch-based techniques and evaluate structure / texture consistency in restored regions.

Unit – I **(11 hours)**
Degradation Models and Evaluation: Image degradation, common noise models, motion / defocus blur intuition, and JPEG characteristics. Controlled experiments using synthetic degradation, train / val / test splits for restoration, and evaluation using PSNR / SSIM.

Unit – II **(11 hours)**
Denoising Methods: Spatial filters, wavelet / transform, non-local means concept. Parameter tuning and over-smoothing control, detail preservation checks, color denoising pitfalls.

Unit – III **(12 hours)**
Deblurring and Deconvolution: Wiener filtering, Richardson–Lucy, regularization, handling ringing artifacts. Blind deconvolution, motion blur estimation heuristics, and robustness to noise and compression.

Unit – IV **(11 hours)**
Inpainting, Super-Resolution and Artifact Removal: Patch-based inpainting, seam and edge consistency. Learning-based inpainting / super-resolution, JPEG artifact reduction, restoration workflow.

Essential Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson)
2. Introduction to Inverse Problems in Imaging by M. Bertero & P. Boccacci (CRC Press)
3. Computer Vision: Algorithms and Applications by Richard Szeliski (Springer)

Suggested Readings

1. Digital Image Processing by William K. Pratt (Wiley)
2. Image Processing: The Fundamentals by Maria Petrou and Panagiota Bosdogianni (Wiley)
3. The Image Processing Handbook by John C. Russ (CRC Press)
4. Image Processing and Analysis by T. Chan & J. Shen (SIAM)

List of Experiments (Hardware / Software) **(30 hours)**

1. Simulation of image degradation models.
2. Image restoration using spatial domain filters.
3. Frequency domain restoration and Wiener filtering.
4. Total variation-based image restoration.
5. Image inpainting using exemplar-based methods.
6. Performance evaluation of restoration and inpainting techniques.

(Note: Course instructor may add / delete / update new experiments in addition to the above suggested practical exercises.)

Department of Electrical Engineering
Faculty of Technology
University of Delhi
Detailed Course Structure and Curriculum of B.Tech. (EE) Fourth Year

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Course Structure of B. Tech. (EE) Fourth Year

Semester VII						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-19	Electrical Drives	3	0	1	4
2	DSE-05	Select a course from the specified list of DSE-05				4
3	DSE-06/ GE-07	Select a course from the specified list of DSE-06/ GE-07				4
4	DSE-07/ GE-08	Select a course from the specified list of DSE-07/ GE-08				4
5	Dissertation on Major OR Dissertation on Minor OR Academic project/ Entrepreneurship					6
					Total Credits	**22**

Semester VIII						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-20	Power Quality and FACTS	3	0	1	4
2	DSE-08	Select a course from the specified list of DSE-08				4
3	DSE-09/ GE-09	Select a course from the specified list of DSE-09/ GE-09				4
4	DSE-10/ GE-10	Select a course from the specified list of DSE-10/ GE-10				4
5	Dissertation on Major OR Dissertation on Minor OR Academic project/ Entrepreneurship					6
					Total Credits	**22**

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week.
T (01 Credit) is equivalent to 01 contact hour per week.
P (01 Credit) is equivalent to 02 contact hours per week.

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department of Electrical Engineering in Fourth Year

S. No.	Semester	DSE	DSE
1.	VII	DSE-05	Advanced Power System Analysis
2.			Cyber Security for Smart Grid
3.		DSE-06	High Voltage Engineering
4.			Power System Optimization
5.		DSE-07	Advanced Power Converters
6.			Modern Distribution System Design
7.	VIII	DSE-08	HVDC Transmission
8.			Net Zero Building Design
9.		DSE-09	Generalized Machine Theory
10.			Power System Security
11.		DSE-10	Analytical Instrumentation
12.			Biomedical Instrumentation

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Specializations and Minor offered by the Department of Electrical Engineering

<table>
<tr><th rowspan="2">S. No.</th><th rowspan="2">Sem</th><th rowspan="2">GE</th><th rowspan="2">Minor in EE (Open only for CSE/ECE)</th><th colspan="3">Specializations for EE / Minors for ECE/CSE</th></tr>
<tr><th>Robotics & Automation</th><th>Sustainable Energy Engineering</th><th>Electric and Hybrid Vehicle</th></tr>
<tr><td>1</td><td>VII</td><td>DSE-06/ GE-07</td><td>Element of Electrical Power System</td><td>Nonlinear Control System</td><td>Machine Learning Applications in Sustainable Energy Development</td><td>Electro-Chemistry of Fuel Cells</td></tr>
<tr><td>2</td><td>VII</td><td>DSE-07/ GE-08</td><td>Control System Applications in Electrical Engineering</td><td>Digital Robotic Control</td><td>Control of Renewable Energy Integrated Systems</td><td>Modeling and Simulation of EHV</td></tr>
<tr><td>3</td><td rowspan="2">VIII</td><td rowspan="2">DSE-09/ GE-09</td><td>Power Plant Engineering</td><td>Real Time Embedded System</td><td>Smart Energy Storage System</td><td>Embedded Based Smart System Design</td></tr>
<tr><td>4</td><td>Machine Learning Applications in Electrical Engineering</td><td>Process Control</td><td>Wind and Small Hydro Power Generation</td><td>Computer Aided Design of Electric Machines</td></tr>
<tr><td>5</td><td rowspan="2">VIII</td><td rowspan="2">DSE-10/ GE-10</td><td>Clean Energy Technologies</td><td>Optimal and Robust Control</td><td>Smart Grid Technologies</td><td>EV Charging Infrastructure</td></tr>
<tr><td>6</td><td>Introduction to Smart Grid</td><td>Robotics in Biomedical Engineering</td><td>Energy Management Systems and SCADA</td><td>Electrical Vehicle System Design</td></tr>
</table>

Detailed Syllabus of Discipline Specific Core (DSC) Courses of B. Tech. (EE) – SEMESTER VII

Electrical Drives (DSC-19)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electrical Drives	4	3	0	1	Power Electronics, Electrical Network Analysis, Electric Machine I

Course Objectives:

1. Describe the components of the Electric Drives System and their choice criterion.
2. Analyze the steady state and transient stability of Electric Drives
3. Understand the various control techniques for electric drive systems.
4. Select the particular motor according to the duty it is used in drives
5. Comprehend the basic concepts of DC & AC drives and know their characteristics.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:
1. After studying the basic concepts of electric drives.
2. Understanding of power circuit configuration of the phase-controlled rectifiers and choppers for the speed control of DC motor drives.
3. Design static Scherbius and Kramer drives to implement slip power recovery schemes
4. Design and implementation of synchronous motor drives with fixed frequency and variable frequency sources

UNIT-I ***(13 hours)***

Fundamentals Of Electric Drives

Electrical drives and introduction: Electric drives, advantages of electrical drives, parts of electrical drives, and choice of electrical drives. Types of loads, load with translational motion, load with rotational motion, load torque that varies with time, Selection of Motor Power Rating.

Dynamics Of Electrical Drives

Fundamental torque equation, speed-torque convention, and multi-quadrant operation, dynamics of motor load combination, nature and classification of load torque, measurement of moment of inertia, calculation of acceleration time in transient operation, acceleration time for specific nature of motor and load torque, load equalization, stability of electrical drives

UNIT-II ***(12 hours)***

Rectifier control of DC drives- separately excited DC motor drives using controlled rectifiers single-phase fully controlled rectifier fed drives (discontinuous and continuous mode of operation), critical speed - single-phase semi converter fed drives (continuous mode of operation) - three-phase semi converter and fully controlled converter fed drives (continuous mode of operation) - dual converter control of DC motor - circulating current mode.

Chopper control of DC drives - two-quadrant and four-quadrant chopper drives - motoring and regenerative braking - chopper-fed DC series motor drive - closed-loop speed control for separately excited DC motor.

UNIT-III ***(08 hours)***

Three phase induction motor drives: Stator voltage control - Stator frequency control – v/f control - below and above base speed – Voltage Source Inverter (VSI) fed v/f control using sine-triangle PWM - static rotor resistance speed control employing chopper – static slip power recovery speed control scheme for speed control below synchronous speed.

UNIT-IV ***(12 hours)***

Concept of space vector – Clarke and Park transformation – field orientation principle – Introduction to direct vector control of induction motor drives – decoupling of flux and torque components - space vector diagram and block diagram**.**

Three Phase Synchronous Motors

Review of Three phase Synchronous Motor and its performance, self-controlled schemes, Variable frequency control of multiple synchronous motor, Permanent magnet AC motor drives, Brushless DC Motor Drives

Essential Readings:

1. G. K. Dubey, *Fundamentals of Electrical Drives*, Narosa Publishing House (2nd Edition, 2008 or later).
2. R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control*, Pearson Education.
3. Werner Leonhard, *Control of Electrical Drives*, Springer (3rd Edition, 2001).
4. Vedam Subrahmanyam, *Electrical Drives: Concepts and Applications*, McGraw Hill / TMH.
5. B. K. Bose, *Modern Power Electronics and AC Drives*, Pearson Publication.

Suggested Readings:

1. P. C. Sen, *Principles of Electric Machines and Power Electronics*, John Wiley & Sons, 3rd Edition.
2. M. H. Rashid, *Power Electronics: Circuits, Devices, and Applications*, Prentice Hall / Pearson.
3. N. Mohan, T. M. Undeland, and W. P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, 3rd Edition.
4. Bin Wu and Mehdi Narimani, *High-Power Converters and AC Drives*, Wiley-IEEE Press, 2nd Edition, 2017.
5. P. Vas, *Sensorless Vector and Direct Torque Control*, Oxford University Press.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Study of speed control of Thysistor controlled DC Drive.
2. Study of speed control of Chopper-fed DC Drive
3. Experimental studies on different types of braking of separately excited DC motor. (a) Dynamic braking (b) Plugging
4. Study of PWM Inverter fed 3 phase Induction Motor control.
5. Study of VSI / CSI fed Induction motor Drive.
6. Study of V/f control of 3phase Induction motor drive.
7. Study of permanent magnet synchronous motor drive fed by PWM Inverter.
8. Study of Regenerative / Dynamic braking operation for DC Motor.
9. Study of Regenerative / Dynamic braking operation of AC motor.
10. Study of PC/PLC based AC/DC motor control operation.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (EE) – SEMESTER VII

Advanced Power System Analysis (DSE-05)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Advanced Power System Analysis	4	3	0	1	Power System Analysis

Course Objectives:

1. To analyze a Power System Network using graph theory.
2. To interpret the formation of Network matrices.
3. To construct the necessity of load flow studies and various methods of Analysis.
4. To understand the contingency.
5. To formulate the power system matrix for an unbalanced system.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Formulate Y_{BUS} and Z_{BUS} matrices for power network analysis, including modifications and coupled branches.
2. Apply load flow methods (Gauss-Seidel, N-R, decoupled, DC) to solve transmission and load flow for distribution systems.
3. Perform contingency analysis for single/multiple outages using piecewise and reduction techniques.
4. Analyze three-phase unbalanced systems and faults in the phase domain.

UNIT-I ***(15 hours)***

Introduction:

Admittance Model and Network Calculations, Branch and Node Admittances, Mutually Coupled Branches in Y_{BUS}, An Equivalent Admittance Network, Modification of Y_{BUS}, Network Incidence Matrix and Y_{BUS}, Sparsity and Near Optimal Ordering. Impedance Model and Network Calculations, the Bus Admittance and Impedance Matrices, Thevenin's Theorem and Z_{BUS}, Algorithms for building Z_{BUS}, Modification of existing Z_{BUS}, Calculation of Z_{BUS} elements from Y_{BUS}, Power Invariant Transformations, Mutually Coupled Branches in Z_{BUS}.

UNIT-II ***(12 hours)***

Computer-based load flow analysis: Gauss-Seidel method, N-R Method, Decoupled method, fast decoupled method, comparison between power flow solutions. DC load flow. Load flow for distribution systems.

UNIT-III ***(10 hours)***

Contingency Analysis, Adding and Removing Multiple Lines, Piecewise Solution of Interconnected Systems, Analysis of Single Contingencies, Analysis of Multiple Contingencies, System Reduction for Contingency and Fault Studies.

UNIT-IV ***(08 hours)***

Formulation of the Y_{BUS} matrix of a three-phase unbalanced system, Fault Analysis: Fault analysis in the phase domain.

Essential Readings:

1. John J. Grainger and W. D. Stevenson, "Power System Analysis", McGraw-Hill Education.
2. I. J. Nagrath and D. P. Kothari, "Modern Power System Analysis", McGraw Hill Publishing Company Ltd, 2nd Edition.
3. J. Duncan Glover and M.S. Sarma, "Power System Analysis and Design", Nelson Engineering.
4. Abhijit Chakrabarti and Sunitha Halder, "Power Systems Analysis, operation and control", PHI Learning Pvt. Ltd.

Suggested Readings:

1. Olle. L. Elgerd, "Electrical Energy Systems Theory", McGraw Hill Education.
2. Prabha Kundur, "Power Systems Stability and Control", McGraw-Hill Publishing Company Ltd.
3. K. Uma Rao, "Power System Operation and Control", Wiley India Pvt. Ltd.
4. P.S.R. Murthy, "Operation and Control in Power Systems", BS Publications.
5. Robert H. Miller and Jamesh H. Malinowski, "Power System Operation", McGraw-Hill Professional Pub.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Build Y_{BUS} from branch/node data and verify sparsity using the incidence matrix.
2. Construct Z_{BUS} by building algorithms and computing elements from YBUS.
3. Implement Gauss-Seidel load flow and compare convergence with Newton-Raphson, decoupled, and fast decoupled methods for large system power flow.
4. Develop a DC load flow model for quick active power assessment.
5. Analyze single contingencies by line removal and performance metrics.
6. Evaluate multiple contingencies with piecewise interconnected solutions.
7. Formulate a three-phase YBUS and perform phase-domain fault analysis.
8. Formulate a load flow for unbalanced systems.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Cyber Security for Smart Grid (DSE-05)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cyber Security for Smart Grid	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. Introduce students to the architecture, operation, and evolution of the smart electric grid as a cyber-physical system.
2. Provide fundamental knowledge of sensing, communication, and control technologies used in smart grids.
3. Familiarize students with basic cybersecurity concepts, threats, and vulnerabilities in smart grid environments.
4. Enable students to understand common cyber-attacks and basic protection mechanisms used in power system applications.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the operational structure and key components of a smart grid.
2. Describe sensing, communication, and networking principles used in smart grid systems.
3. Identify major smart grid communication and control protocols used in power systems.
4. Analyze basic cybersecurity threats and vulnerabilities affecting smart grid infrastructure.
5. Understand fundamental security mechanisms and real-world cyber-attack case studies in smart grids.

UNIT-I ***(13 hours)***

Smart Grid Fundamentals: Overview of the conventional electric power grid and its operation, causes of failures and disturbances in power grid operation, learning from historical blackout events and grid failures, evolution toward the smart grid.

Cyber-Physical Systems: Smart grid architecture and its classification as a cyber-physical system, key components of a smart grid: sensors, control systems, and information and communication technologies, introduction to cybersecurity challenges in smart grids, organizational structures for ensuring system reliability, role of standards and interoperability.

UNIT-II ***(12 hours)***

Sensing: Sensing and measurement in smart grids, introduction to Phasor Measurement Units (PMUs): need, basic working principle, applications, and time synchronization, smart meters and their role in advanced metering infrastructure.

Communication and Networking: Communication infrastructure in smart grids, communication performance parameters: propagation delay, transmission delay, queuing delay, jitter, data loss, and corruption, layered communication models: OSI and TCP/IP, overview of physical, data link, network, transport, and application layers with relevance to smart grid communication.

UNIT-III ***(10 hours)***

Smart Grid Communication Protocols: Introduction to power system application-layer protocols, SCADA systems and their importance in monitoring and control, overview of SCADA communication protocols: DNP3 and IEC 61850, Inter-Control Center Communication Protocol (ICCP), synchro phasor communication standard IEEE C37.118, communication requirements for smart metering and distributed energy resources.

Control Infrastructure: Conventional and modern control centers, introduction to fault-tolerant computing, basic concepts of cloud computing in smart grid control centers.

UNIT-IV ***(10 hours)***

Cybersecurity Fundamentals, Attacks, and Protection in Smart Grid: Basic cybersecurity concepts: threats, vulnerabilities, and risks in power systems, CIA triad: confidentiality, integrity, and availability, privacy concerns and consumer data protection in smart grids, introduction to cryptography concepts: encryption, authentication, symmetric and asymmetric encryption, digital signatures, and certificates, overview of cybersecurity standards such as IEC 62351 and NERC CIP, common network attacks on smart grids: Denial-of-Service, spoofing, and malware, basic protection mechanisms: firewalls, intrusion detection systems, and secure communication protocols, case studies of major cyber-attacks on power grids including Stuxnet and the Ukraine power grid attack.

Essential Readings:

1. A.K. Srivastava, V. Venkataramanan, C. Hauser, Cyber Infrastructure for the Smart Electric Grid, Willey-IEEE Press.
2. A. Sreejith, K.S. Swarup, Cyber-Security for Smart Grid Control: Vulnerability Assessment, Attack Detection, and Mitigation, Transactions on Computer Systems and Networks, Springer.
3. S. Sridhar, A. Hahn, and M. Govindarasu, Cyber-Physical System Security for the Smart Grid, Elsevier.

Suggested Readings:

1. Eric D. Knapp and Joel T. Langill, Industrial Network Security, Syngress.
2. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To study the architecture and components of a smart grid using simulation tools.
2. To model a basic power distribution system and observe its operation under normal conditions.
3. To study the working and applications of Phasor Measurement Units (PMUs).
4. To analyze communication delay, jitter, and packet loss in a simple smart grid communication network.
5. To study OSI and TCP/IP protocol layers using network simulation tools.
6. To simulate basic SCADA communication using DNP3 or IEC 61850 concepts.
7. To analyze the impact of Denial-of-Service (DoS) attacks on smart grid communication networks.
8. To demonstrate spoofing attacks and their effects on network communication.
9. To study basic encryption and authentication techniques for securing smart grid data.
10. To implement a simple firewall or intrusion detection mechanism for smart grid communication.
11. To analyze real-world smart grid cyber-attack case studies and propose basic mitigation measures.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

High Voltage Engineering (DSE-06)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
High Voltage Engineering	4	3	0	1	Mathematics-I, Electromagnetic Field Theory

Course Objectives:

1. To know about generation voltages and currents to test the electrical equipment.
2. To understand the general aspects of electrical testing methods.
3. To understand the need for testing and procedures of the electrical devices.
4. To understand the need for high voltages, high currents, and transients.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand the general aspects of electrical testing methods.
2. Analyze various generation and measurement techniques for high Voltages and Currents.
3. Identify the overvoltages in power systems and outline the principles of insulation coordination for various parts of the power system.
4. Analyze various generation and measurement techniques for high Voltages and Currents.

UNIT-I ***(10 hours)***

Introduction to High Voltage Engineering: Electric field stresses. Gas/Vacuum as an insulator, Liquid dielectrics. Solids and composites, Estimation and control of Electric stress, Numerical methods for electric field computation. Surge voltages, their distribution and control, Applications of insulation materials in transformers, rotating machines, circuit breakers, cables, power capacitors, and bushings.

UNIT-II ***(11 hours)***

Generation of High Voltages: Generation of high direct voltages, half and full wave rectifier circuits, voltage multiplier circuits, Van de Graff generators, electrostatic generators, examples - generation of alternating voltages, testing transformers, cascaded transformers, resonant transformers, examples - impulse voltages, Standard lightning and switching surge and associated parameters and their corrections, impulse voltage generator circuits, Marx circuit, operation, design and construction of impulse generators, examples - impulse current generator - control systems.

UNIT-III ***(12 hours)***

Measurement and Testing: Measurement of high voltages: Sphere gaps, factors affecting sphere gap measurements, correction factors. Measurement of high AC voltage: Capacitance voltage dividers, Chubb-Fortescue method, CVT, electrostatic voltmeters. Measurement of high DC voltage: Resistive voltage dividers, Generating voltmeter. Measurement of impulse voltage: Capacitance divider, Impedance matching. Measurement of transient currents: Resistive shunt, Magnetic coupling, Hall Effect current transducers, Integrating and differentiating type Rogowski coils. Digital techniques in HV measurements, DSO.

UNIT-IV ***(12 hours)***

Insulation Materials and Breakdown: Introduction to solid, liquid and gaseous insulators used in power equipment. Classifications of insulation based on temperature withstand limits, dielectric losses, ageing of insulation materials (paper–press board) and remaining life analysis. Applications of nanofilled materials for outdoor and indoor insulation. Introduction to solid, liquid and gaseous dielectrics. Breakdown in gas and gas mixtures-breakdown in uniform and non-uniform fields-Paschen’s law-Townsends criterion-streamer mechanism-corona discharge-breakdown in electro negative gases. Breakdown in liquid dielectrics-suspended particle mechanism. Breakdown in solid dielectrics - intrinsic, streamer and thermal breakdown.

Essential Readings:

1. Kuffel and Zaengl, High Voltage Engineering Fundamentals, 2nd ed., Newness, 2002.
2. M. S. Naidu, V. Kamaraju, High Voltage Engineering, 3rd ed., McGraw-Hill,1995.
3. M. Khalifa, High Voltage Engineering: Theory and Practice, Dekker, 1990.

Suggested Readings:

1. H. M. Ryan, High Voltage Engineering and Testing, IEE 2001.
2. Wadhwa C L, High Voltage Engineering, New Age International, New Delhi,1994.

Practical Component: ***(30 hours)***

List of Experiments:

1. AC, DC and impulse breakdown test of solid insulation.
2. Oil breakdown test using oil test kit.
3. Capacitance and tanʒ measurement of insulator.
4. Dielectric characteristics of solid insulating material using impedance analyzer.
5. Measurement of insulation resistance of cable.
6. Mapping of electric field lines between two charges using MATLAB
7. Simulation of dc/impulse voltage generation circuits using PSPICE/PSCAD.
8. Preparation of epoxy nanocomposite plotting the electrical field distribution in an insulating material using COMSOL (with and without void).
9. Field distribution between plates of a parallel plate capacitor using ANSYS and COMSOL using Finite Element Method.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Power System Optimization (DSE-06)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power System Optimization	4	3	0	1	Power System Analysis

Course Objectives:

1. Understand feed-forward neural networks, feedback neural networks, and learning techniques.
2. Understand Non-linear problems.
3. Understand the concept of metaheuristic algorithm.
4. Develop genetic algorithm for applications in electrical engineering.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain linear and non-linear optimization methods, including basic LP and NLP techniques.
2. Describe feed-forward and feedback neural networks and related AI concepts.
3. Formulate and solve multistage decision problems using dynamic programming.
4. Design and apply genetic algorithms for constrained and unconstrained optimization problems.
5. Use classical methods and genetic algorithms to solve and interpret key power system problems such as load flow, ELD, unit commitment, reactive power optimization, and OPF.

UNIT-I ***(10 hours)***

Introduction:

Introduction to Artificial Neural Networks, Artificial Intelligence and Neural networks, Introduction to optimization and classical optimization techniques. Linear Programming: Standard form, geometry of LPP, Simplex Method pf solving LPP, revised simplex method, duality, decomposition principle, and transportation problem.

UNIT-II ***(08 hours)***

Non-Linear Problem (NLP):

One dimensional method, Elimination methods, Interpolation methods, Unconstrained optimization techniques-Direct search and Descent methods, constrained optimization techniques, direct and indirect methods.

UNIT-III ***(12 hours)***

Dynamic Programming:

Multistage decision processes, concept of sub-optimization and principle of optimality.

Genetic Algorithm:

Introduction to genetic Algorithm, working principle, coding of variables, fitness function. GA operators; Similarities and differences between Gas and traditional methods; Unconstrained and constrained optimization using Genetic Algorithm, real coded gas, Advanced Gas, global optimization using GA.

UNIT-IV ***(15 hours)***

Applications to Power System-I:

Load flow studies, Economic Load Dispatch in thermal and Hydro-thermal system using GA and classical optimization techniques.

Applications to Power System-II:
Unit commitment problem, reactive power optimization. Optimal power flow, LPP and NLP techniques to optimal flow problems.

Essential Readings:
1. S. S.Rao, "Optimization - Theory and Applications", Wiley-Eastern Limited.
2. David G. Luenberger, "Introduction of Linear and Non-Linear Programming", Addison Wesley Publishing Company.
3. E. Polak, "Computational methods in Optimization", Academic Press Inc.
4. L.P. Singh, "Advanced Power System Analysis and Dynamics", New Age International Private Limited.

Suggested Readings:
1. Donald A. Pierre, "Optimization Theory with Applications", Dover Publications.
2. Kalyanmoy Deb, "Optimization for Engineering Design: Algorithms and Examples", PHI Publication.
3. D. E. Goldberg, "Genetic Algorithm in Search Optimization and Machine Learning", Addision-Wesley Publication.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To formulate and solve linear programming problems using the simplex method and compare manual and software-based solutions for selected electrical engineering cases.
2. To implement and analyze one-dimensional and unconstrained non-linear optimization methods (e.g., line search / direct search) for standard test functions relevant to engineering problems.
3. To model a multistage decision problem and apply dynamic programming to demonstrate the principle of optimality for a simple resource allocation or scheduling task.
4. To design and implement a basic genetic algorithm, including coding of variables, fitness function, and genetic operators, for solving an unconstrained optimization problem.
5. To apply genetic algorithms and classical optimization techniques for Economic Load Dispatch of a thermal power system and compare their performance in terms of cost and convergence.
6. To formulate and solve the unit commitment or optimal power flow problem using suitable optimization techniques (LPP/NLP or GA) and analyze the impact of constraints on system operation.
7. To model and train a simple feed-forward neural network for an engineering function approximation or classification task, and study the effect of learning parameters on performance.
8. To apply optimization techniques and genetic algorithms to reactive power optimization or voltage profile improvement in a small test power system and evaluate the resulting system performance.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Advanced Power Converters (DSE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Advanced Power Converters	4	3	0	1	Power Electronics, Electrical Network Analysis, Electric Machine

Course Objectives:

This course covers

1. Advanced converter topologies.
2. Resonant and dual-active bridge converters.
3. Impedance source converters.
4. Power electronics applications in high-performance systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. To understand isolated and non-isolated converter topologies, resonant converters, and advanced DAB configurations.
2. To analyze converter modeling, impedance source converters, and control strategies for high-performance applications.
3. To design advanced power converters, including resonant and multilevel impedance source converters.
4. To evaluate converter efficiency, power flow control, and system performance under various operating conditions.

UNIT-I ***(13 hours)***

Introduction: Basics isolated and non-Isolated converter topologies, advantages and disadvantages, state- space and small signal modeling of the basic converter topology, Introduction to advanced converter topologies such as multilevel converter, resonant converter, dual active bridge, matrix converter, overview of power electronics converters application.

Higher Order Converters and Mult quadrant Operation - Boost Circuit, Two- Stage, Three-Stage, Higher Stage Boost Circuit, Introduction to double and triple Boost Circuit; Circuit Description, Discontinuous Region, Switched Component Converters, Switched Capacitors DC/DC Converters, Switched Inductor Four-Quadrant DC/DC Converter, Continuous Mode, Discontinuous Mode.

UNIT-II ***(12 hours)***

Resonant Converter: Introduction, need of Resonant Power Converter; Resonant Converter Applications; Overview of Resonant, Quasi-Resonant and Zero-Transition Converter Topologies; Design Considerations Rectifiers: Basics and types of Rectifiers; Current Rectifiers; Voltage Rectifiers; Series / Parallel Resonant Inverters: Introduction, Series Resonant Inverters, Parallel Resonant Inverters, Series-parallel Resonant Inverters; Resonant DC-DC Converters: Series Resonant Converters, Parallel Resonant Converters, Series-Parallel Resonant converters, LLC Converters, Bidirectional CLLC Converters.

UNIT-III ***(08 hours)***

Dual Active Bridge Converters: Basic principles and configurations of Dual Active Bridge (DAB) converters; Comparison with other converter topologies; Converter Modeling and Analysis: Small-signal modeling of DAB converters, Analysis of steady-state and transient behavior, Power flow analysis and efficiency optimization techniques; Control Strategies for DAB Converters: Voltage and current control techniques, Phase-shift modulation and its implementation**.**

UNIT-IV ***(12 hours)***

Introduction to Special Machine Drives & Synchronous motors Drive: Construction, operation from fixed frequency supply –starting, speed-torque characteristics, pulling in, braking of motor various special motors such as Synchronous Motor, BLDC, Switch Reluctance Motor, PMSM. Variable speed Drives. Self-controlled synchronous motor drive employing load commutated inverter. brushless dc motor controllers–rotor position measurement, commutation logic, speed controller.

Essential Readings:

1. Bin Wu and Mehdi Narimani, *High-Power Converters and AC Drives*, Wiley-IEEE Press, 2nd Edition, 2017.
2. Ned Mohan, Tore M. Undeland, and William P. Robbins, *Power Electronics: Converters, Applications, and Design*, John Wiley & Sons, 3rd Edition, 2003.
3. M. H. Rashid (Ed.), *Power Electronics Handbook*, Butterworth-Heinemann / Elsevier, 5th Edition, 2017.
4. D. Grahame Holmes and Thomas A. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*, IEEE Press / Wiley, 2003.

Suggested Readings:

1. Robert W. Erickson and Dragan Maksimović, *Fundamentals of Power Electronics*, Springer, 2nd Edition, 2001.
2. Jose Rodriguez and Patricio Cortes, *Predictive Control of Power Converters and Electrical Drives*, Wiley-IEEE Press, 2012.
3. B. K. Bose, *Modern Power Electronics and AC Drives*, Pearson Education, 2002.
4. L. Umanand, *Power Electronics: Essentials and Applications*, John Wiley India, 2009.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Simulation and analysis of basic isolated (flyback, forward) and non-isolated (buck, boost, buck-boost) converter topologies: Compare advantages/disadvantages, efficiency, and ripple using MATLAB/Simulink.
2. State-space modeling and small-signal analysis of a basic boost converter; verify step response and stability.
3. Simulation of higher-order boost converters (two-stage, three-stage, or quadratic boost): Study voltage gain, continuous/discontinuous modes, and comparison with conventional boost.
4. Simulation of switched-capacitor or switched-inductor DC-DC converters; analyze four-quadrant operation (e.g., switched-inductor four-quadrant converter in continuous and discontinuous modes).
5. Overview simulation: Multilevel converter (e.g., cascaded H-bridge or diode-clamped) and basic resonant converter topology; compare with conventional converters for applications like renewables.
6. Simulation of series-parallel resonant inverter and resonant DC-DC converters (series, parallel, and series-parallel types).
7. Design and simulation of LLC resonant converter: Study gain characteristics, soft-switching, and efficiency optimization.
8. Simulation of bidirectional CLLC resonant converter: Analyze power flow in both directions and compare with LLC.
9. Simulation of single-phase Dual Active Bridge (DAB) converter: Implement phase-shift modulation, analyze power flow, steady-state behavior, and efficiency.
10. Small-signal modeling and control of DAB converter: Simulate voltage/current control strategies and transient response.
11. Simulation of self-controlled synchronous motor drive using load-commutated inverter or brushless DC motor (BLDC) controller: Study rotor position sensing, commutation, and speed control.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Modern Distribution System Design (DSE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Modern Distribution System Design	4	3	0	1	Power Transmission and Distribution

Course Objectives:

1. Understand the components of electric power distribution systems.
2. Modelling of different distribution system components.
3. Understand the analysis methods, specially developed for the distribution system.
4. Understand the distribution system protection devices.
5. Understand the components of distribution automation systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand the structure and components of modern electric power distribution systems.
2. Develop basic skills for modeling key distribution system elements such as lines, transformers, regulators, capacitors, and DG units.
3. Grasp and apply analysis methods specifically developed for radial and weakly meshed distribution networks, including load-flow techniques.
4. Understand the principles, operation, and coordination of distribution protection devices.
5. Gain foundational knowledge of distribution automation systems and their role in improving reliability and system performance.

UNIT-I ***(05 hours)***

Structure of a distribution system, Distribution feeder configurations and substation layouts, Nature of loads.

UNIT-II ***(15 hours)***

Approximate methods of analysis, Computation of transformer and feeder loading, “K” Factors, voltage drop and power loss calculations, Distribution of loads, and various geometric configurations.

UNIT-III ***(10 hours)***

Modelling of distribution system components, Overhead lines, feeders and cables, Single and three phase distribution transformers, Voltage regulators, Load models, Capacitor banks, Distributed generation.

UNIT-IV ***(15 hours)***

Distribution system analysis, Load flow analysis:
Backward/forward sweep, Load flow analysis: Direct approach, Load flow analysis: Direct approach for weakly meshed systems.

Distribution system protection

Distribution system protection devices, Problems in distribution systems, and the need for automation.

Essential Readings:

1. Turan Gonen, “Electric Power Distribution System Engineering”, CRC Press.
2. A. S. Pabla, “Electric Power Distribution”, McGraw-Hill Publication.

3. James A Momoh, “Electric Power Distribution, Automation, Protection and Control”, CRC Press.

Suggested Readings:

1. Abdelhay A. Sallam and Om P. Malik, “Electric Distribution Systems”, Wiley Publication.
2. William H. Kersting, “Distribution System Modeling and Analysis”, CRC Press.
3. Subrahmanyam S. Venkata, Anil Pahwa, “Electric Power and Energy Distribution Systems Models, Methods, and Applications”, John Wiley & Sons.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Study of Radial and Ring Feeder Configurations; measure voltage profiles and load currents under balanced/unbalanced residential loads to observe feeder performance.
2. Model single-bus and double-bus substation arrangements; analyze fault isolation and load transfer between feeders with varying load natures.
3. Calculate the voltage drop and power loss in a feeder.
4. Load a single-phase distribution transformer with stepped loads; calculate efficiency, "K" factors, and geometric load distribution while measuring voltage regulation.
5. Measure R, L, and C parameters of overhead lines and underground cables using an LCR meter; simulate capacitance effects and compare voltage profiles with Carson's equations.
6. Implement on-load tap-changing regulators and switched capacitors on a feeder; observe power factor improvement and voltage stability under varying loads.
7. Simulate radial distribution load flow; iterate backward current summation and forward voltage updates for a 33/11 kV feeder with DG integration.
8. Program matrix-based direct method for feeders; compare convergence with backward/forward sweep and analyze voltage profiles in unbalanced systems.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Discipline Specific Core (DSC) courses for B.Tech. (EE) – SEMESTER VIII

Power Quality and FACTS (DSC-20)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power Quality and FACTS	4	3	0	1	Power Systems, Power Electronics, Electrical Machines, Control Systems

Course Objectives:

1. To understand the characteristics of AC transmission and the effect of shunt and series reactive compensation.
2. To understand the working principles of FACTS devices and their operating characteristics.
3. To understand the basic concepts of power quality.
4. To understand the working principles of devices to improve power quality.
5. To solve numerical problems on the topics studied

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Analyse the uncompensated AC transmission line.
2. Explain the working principles of FACTS devices and their operating characteristics.
3. Apply FACTS devices for power flow control and stability.
4. Identify different issues of power quality in the distribution system.
5. Apply different compensation and control techniques for DSTATCOM6. explain the working principle of the dynamic voltage restorer and UPQC.

UNIT-I ***(12 hours)***

Transmission Lines and Series/Shunt Reactive Power Compensation: Basics of AC Transmission. Analysis of uncompensated AC transmission lines. Passive Reactive Power Compensation. Shunt and series compensation at the mid-point of an AC line. Comparison of Series and Shunt Compensation.
Thyristor-based Flexible AC Transmission Controllers (FACTS): Description and Characteristics of Thyristor-based FACTS devices: Static VAR Compensator (SVC), Thyristor Controlled Series Capacitor (TCSC), Thyristor Controlled Braking Resistor and Single Pole Single Throw (SPST) Switch. Configurations/Modes of Operation, Harmonics and control of SVC and TCSC. Fault Current Limiter.

UNIT-II ***(12 hours)***

Voltage Source Converter based (FACTS) controllers: Voltage Source Converters (VSC): Six Pulse VSC, Multi-pulse and Multi-level Converters, Pulse-Width Modulation for VSCs. Selective Harmonic Elimination, Sinusoidal PWM and Space Vector Modulation. STATCOM: Principle of Operation, Reactive Power Control: Type I and Type II controllers, Static Synchronous Series Compensator (SSSC) and Unified Power Flow Controller (UPFC): Principle of Operation and Control. Working principle of Interphase Power FlowController. Other Devices: GTO Controlled Series Compensator. Fault Current Limiter.

UNIT-III ***(11 hours)***

Application of FACTS: Application of FACTS devices for power-flow control and stability improvement. Simulation example of power swing damping in a single-machine infinite bus system using a TCSC. Simulation example of voltage regulation of transmission mid-point voltage using a STATCOM.

Power Quality Problems in Distribution Systems: Power Quality problems in distribution systems: Transient and Steady state variations in voltage and frequency. Unbalance, Sags, Swells, Interruptions, Wave-form Distortions: harmonics, noise, notching, dc-offsets, fluctuations. Flicker and its measurement. Tolerance of Equipment: CBEMA curve.

UNIT-IV ***(10 hours)***

DSTATCOM: Reactive Power Compensation, Harmonics and Unbalance mitigation in Distribution Systems using DSTATCOM and Shunt Active Filters. Synchronous Reference Frame Extraction of Reference Currents. Current Control Techniques for DSTATCOM.

Dynamic Voltage Restorer and Unified Power Quality Conditioner: Voltage Sag/Swell mitigation: Dynamic Voltage Restorer – Working Principle and Control Strategies. Series Active Filtering. Unified Power Quality Conditioner (UPQC): Working Principle. Capabilities and Control Strategies.

Essential Readings:

1. Narain G. Hingorani and Laszlo Gyugyi, *Understanding FACTS: Concepts and Technology of Flexible AC Transmission Systems*, IEEE Press / Wiley-IEEE Press, 1999.
2. K. R. Padiyar, *FACTS Controllers in Power Transmission and Distribution*, New Age International Publishers, 1st Edition, 2007.
3. Bhim Singh, Ambrish Chandra, and Kamal Al-Haddad, *Power Quality: Problems and Mitigation Techniques*, John Wiley & Sons Ltd., 2015.
4. Roger C. Dugan, Mark F. McGranaghan, Surya Santoso, and H. Wayne Beaty, *Electrical Power Systems Quality*, McGraw-Hill Education, 3rd Edition, 2012.

Suggested Readings:

1. R. Mohan Mathur and Rajiv K. Varma, Thyristor-Based FACTS Controllers for Electrical Transmission Systems, IEEE Press / John Wiley & Sons, 2002.
2. Math H. J. Bollen, Understanding Power Quality Problems: Voltage Sags and Interruptions, IEEE Press, 2000.
3. Arindam Ghosh and Gerard Ledwich, Power Quality Enhancement Using Custom Power Devices, Springer, 2012.
4. J. Arrillaga, N. R. Watson, and S. Chen, Power System Quality Assessment, John Wiley & Sons, 2000.

Practical Components: ***(30 hours)***

List of Experiments:

1. The Effect of Nonlinear Loads on Power Quality.
2. Study of the Voltage Sag and Voltage Swell with Passive Load and Starting of Induction Motor.
3. To demonstrate the Voltage & Current Distortions with the LED load.
4. To Reduce the Current Harmonics with Filters.
5. To Study the Capacitor Switching Transients.
6. Study the Current Harmonics in BLDC Motor.
7. Study the Effect of Voltage Flicker.
8. Analysis and Simulation of harmonics for various residential loads.
9. Study the neutral current of unbalanced nonlinear load of a 3-Ph System.
10. Reducing the Voltage Sag and Swell Problem in Distribution System Using DVR with PI Controller.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (EE) – SEMESTER VIII

HVDC Transmission (DSE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
HVDC Transmission	4	3	0	1	Power Transmission and Distribution, Power Electronics

Course Objectives:

1. Appraise the need of HVDC technology for bulk power transmission and choose appropriate type of HVDC link and converter.
2. To understand the harmonics and faults occur in the system and their prevention.
3. To evaluate the operation and efficacy of different controllers and analyze the different faults in HVDC systems.
4. To evaluate the issues related to harmonics, reactive power control and protection of HVDC systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Identify HVDC power terminal equipment, classify type of HVDC connectivity and planning of HVDC system.
2. Interpret different types of converter control techniques.
3. Understand the choice of pulse conversion, control characteristic, firing angle control.
4. Calculate voltage and current harmonics, and design of filters and understand the reactive power necessity of conventional control.

UNIT-I ***(11 hours)***

Introduction and Analysis of HVDC Converter: Comparison of HVAC and HVDC transmission system, Applications of DC transmission, Description of DC transmission system, Configurations, Modern trends in DC transmission.

Choice of converter configuration, Simplified analysis of Graetz circuit, Converter bridge characteristics, Characteristics of a twelve-pulse converter, Detailed analysis of converters with and without overlap.

UNIT-II ***(12 hours)***

HVDC Converter and Reactive Power control: General, Principles of DC link control, Converter control characteristics, System control hierarchy, Firing angle control, Current and extinction angle control, Starting and stopping of DC link, Power control, Higher level controllers.

Reactive power requirements in steady state, Sources of reactive power, Static VAR systems, Reactive power control during transients, Harmonics and filters, Generation of harmonics, Design of AC filters and DC filters.

UNIT-III ***(12 hours)***

Smoothing reactor, DC line and Fault Protection: Introduction, Smoothing reactors, DC line, Transient over voltages in DC line, Protection of DC line, DC breakers, Monopolar operation, Effects of proximity of AC and DC transmission lines.

Converter faults, Protection against over-currents, Over voltages in a converter station, Surge arresters, Protection against over-voltages.

UNIT-IV ***(10 hours)***

Component Models for the Power Flow analysis of AC/DC Systems: General, Converter model, Converter control, Modelling of DC network, Modelling of AC networks. Modelling of DC links, Solution of DC load flow, Discussion, Per unit system for DC quantities.

Essential Readings:

1. Kimbark, E.W., 'Dircct Current Transmission-vol.1', Wiley Inter science, New York, 1971.
2. Padiyar, K.R.,'HVDC transmission systems', New Age Publishers, 2017.
3. Kamakshaiah, S and Kamaraju, V, 'HVDC Transmission', IT Edition, Tata McGraw Hill Education (India), New Delhi 2017.

Suggested Readings:

1. Arrilaga, J., 'High Voltage Direct Current Transmission', 2nd Edition, Institution of Engineering and Technology, London, 1998.
2. Vijay K. Sood, 'HVDC and FACTS Controllers', Kluwer Academic Publishers, New York, 2013.

Practical Components: ***(30 hours)***

List of Experiments:

1. Study of various HVDC transmission system components and its applications.
2. Study of AC/DC side voltage and current waveforms of six pulse converter system under variable RL Load using simulation.
3. Study of AC/DC side voltage and current waveforms of twelve pulse converter system under variable R-L Load using simulation.
4. Study of reactive power control in HVDC transmission system.
5. Study of various types of Multi terminal HVDC transmission system.
6. Some simulation practices based on HVDC power and voltage stability.
7. Study of DC link control in VSC based HVDC transmission system.
8. Study of various passive filters used in LCC based HVDC transmission system.
9. Operation of VSC for power factor correction at AC side of HVDC system using sinusoidal pulse width modulation.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Net Zero Building Design (DSE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Net Zero Building Design	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. Introduce global building energy scenarios, nZEB principles, and green rating systems.
2. Analyze energy usage, modeling, load calculations, and comfort factors in buildings.
3. Explore renewable integrations like BiPV, storage, and grid-interactive systems.
4. Examine smart buildings, BAS, and energy auditing practices.
5. Foster skills in sustainable design for net-zero and positive-energy buildings.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Define nZEB, EPI, site/source energy, and apply green ratings.
2. Model building envelopes using RC networks and calculate HVAC loads.
3. Evaluate renewables and storage for building energy balance.
4. Assess comfort metrics and passive/active strategies.
5. Implement BAS concepts and conduct basic energy audits.

UNIT-I ***(12 hours)***

Building Energy Consumption: Global Energy and Environmental Scenario, Building Energy Consumption, What is nZEB? Why Zero? Energy Units & Conversions, Definitions, Site and Source Energy, Energy Basis vs. GHG Basis, Building Energy Systems, Energy Performance Indices. Concept of Green Buildings and Ratings.

UNIT-II ***(18 hours)***

Energy Usage in Buildings: Energy use intensity, End uses, Energy transfer processes in building space, Building energy modelling and simulation: Building envelope model-Thermal network model, Resistance-Capacitance (RC) network, transfer function method, single zone model of building space, Heating/cooling load calculation HVAC and Lighting systems, Principles for Net Zero Building Enclosures, Passive and Active Systems.

Occupancy Comfort Analysis and Evaluation: Interactions between forms of comfort and building energy use, Including Thermal comfort, Visual comfort and daylighting, indoor air quality, acoustic comfort, and occupant comfort calculations.

UNIT-III ***(15 hours)***

Renewable Energy for Buildings: Review of renewable energy technologies, BiPV, BiPV/T, BaPV, Solar thermal systems with storage, Solar air-conditioning, combined heat and power, Energy storage: Batteries-Lithium, Lead-acid, Thermal energy storage-Phase change materials, hydrogen. Geothermal. Smart buildings: Grid-interactive buildings, Energy Management, Electric vehicles, Vehicle-Building-Grid integration, smart homes, and positive energy buildings.

UNIT-IV ***(10 hours)***

Introduction to Smart Buildings, Building Automation Systems, Case Studies in Smart Building Implementation, and Introduction to Energy Auditing.

Essential Readings:

1. M.S. Sodha and N.K. Bansal, "Solar Passive: Building Science and Design", Pergamon Press.
2. M. H. Chilogioji and E. N. Oura, "Energy Conservation in Commercial and Residential Buildings," Marcel Dekker Inc., New York.
3. F. S. Dubin and C. G. Long, "Energy Conservation Standards for Building Design," Construction and Operation McGraw Hill.

Suggested Readings:

1. M. Majumdar, "Energy efficient Buildings in India," Tata Energy Research Institute, Ministry of Non Conventional Energy Sources, 2002.
2. J. Khazaii, "Energy-Efficient HVAC Design: An Essential Guide for Sustainable Building," Springer 2014.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Verify site vs source energy conversions using sample data.
2. Simulate RC thermal network for single-zone building model.
3. Calculate EUI and EPI from building consumption records.
4. Analyze thermal comfort via PMV/PPD for indoor conditions.
5. Measure daylighting levels and visual comfort metrics.
6. Test BiPV output under varying irradiance conditions.
7. Evaluate PCM storage for thermal load shifting.
8. Demonstrate BAS control on mock HVAC/lighting setup.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Generalized Machine Theory (DSE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Generalized Machine Theory	4	3	1	0	Electrical Machines I, Electrical Machines II

Course Objectives:

1. Express the revolving field and reference frame theory.
2. Develop mathematical models of three-phase AC machines and parameters in different reference frames.
3. Simulate the transient performance of three-phase AC machines in different reference frames.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Develop the basic two pole model representation of electrical machines using the basic concepts of generalized theory.
2. Apply linear transformation for the steady state and transient analysis of different types of rotating electrical machines.
3. Understand the linear transformation equations of rotating electrical machines incorporating the concept of power invariance.

UNIT-I ***(15 hours)***

Introduction and Reference Frame Transformations:

Unified approach to the analysis of electrical machine performance - per unit system - Basic two pole model of rotating machines- Primitive machine -Conventions -transformer and rotational voltages in the armature voltage and torque equations, resistance, inductance and torque matrix.

Passive linear transformation in machines-invariance of power transformation from a displaced brush axis-transformation from three phase to two phase and from rotating axes to stationary axes-Physical concept of Park's transformation

UNIT-II ***(10 hours)***

DC Machines:

Application of generalized theory to separately excited DC generator: steady state and transient analysis, Separately excited DC motor- steady state and transient analysis, Transfer function of separately excited DC generator and motor- DC shunt and series motors: Steady state analysis and characteristics.

UNIT-III ***(12 hours)***

Synchronous Machines:

synchronous machine reactance and time constants-Primitive machine model of synchronous machine with damper windings on both axes. Balanced steady state analysis-power angle curves.

Induction Machines:

Primitive machine representation. Transformation- Steady state operation-Equivalent circuit. Torque slip characteristics.

UNIT-IV ***(08 hours)***

Single phase induction motor:

Revolving Field Theory equivalent circuit- Voltage and Torque equations-Cross field theory-Comparison between single phase and poly phase induction motor.

Essential Readings:

1. Charles Concordia, Synchronous Machines- Theory and Performance, John Wiley and Sons Incorporate, New York.1988.
2. Say M. G., Introduction to Unified Theory of Electrical Machines, Pitman Publishing, 1978.
3. Alexander SLangsdorf, Theory of Alternating Current Machinery, Tata McGraw Hill, 2nd revised edition, 2001.

Suggested Readings:

1. Mukhopadhyay A. K., Matrix Analysis of Electrical Machines, New Age, 1996.
2. Vas P., Electrical Machines and Drives: A Space-Vector Theory Approach (Monographs in Electrical and Electronic Engineering), Oxford University Press, 1993.

Tutorial Component: ***(15 Hours)***

(Note: The course instructor will design tasks to complete the tutorial component of the course.)

Power System Security (DSE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power System Security	4	3	1	0	Power Transmission and Distribution, Power System Analysis

Course Objectives:

1. Understand the concept of power system security and control.
2. Understand the economic operation of the power system.
3. Load frequency control.
4. Understand the voltage and reactive power control.
5. Understand the concept of State Estimation.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the structure, control centre functions, SCADA, operating states, and security of a modern power system.
2. Model and analyze single-area and two-area load frequency control schemes, including PI/AGC, for steady-state and dynamic performance.
3. Describe and model automatic voltage regulators and excitation systems, and evaluate methods of voltage and reactive power control using transformers and reactive compensation.
4. Discuss and compare methods of voltage and reactive power management, such as shunt, series, and phase-angle compensation in transmission networks.

UNIT-I ***(08 hours)***

Introduction:

Structure of power systems, Power system control center and real time computer control, SCADA system, Level decomposition in power system, Power system security, Various operational stages of power system, Power system voltage stability.

UNIT-II ***(15 hours)***

Load Frequency Control:

Concept of load frequency control, Load frequency control of single area system: Turbine speed governing system and modelling, block diagram representation of single area system, steady state analysis, dynamic response, control area concept, P-I control, load frequency control and economic dispatch control. Load frequency control of two area system: Tie line power modelling, block diagram representation of two area system, static and dynamic response.

UNIT-III ***(15 hours)***

Automatic Voltage Control:

Schematic diagram and block diagram representation, different types of Excitation systems & their controllers.

Voltage and Reactive Power control: Concept of voltage control, methods of voltage control-control by tap changing transformer. Shunt Compensation, series compensation, phase angle compensation.

UNIT-IV ***(07 hours)***

State Estimation:
Detection and identification, Linear and non-linear models.

Essential Readings:

1. P.S.R. Murty, “Operation and control in Power Systems”, BS Publications.
2. Allen J. Wood, Bruce F. Wollenberg and Gerald B. Sheblé, “Power Generation, Operation and Control”, John Wiley & Sons.
3. O.I. Elgerd, “Electric Energy System Theory”, Tata McGraw-Hill Publishing Company Limited.

Suggested Readings:

1. I. J. Nagrath and D. P. Kothari, “Modern Power System Analysis”, McGraw Hill Publishing Company Ltd.
2. Prabha Kundur, “Power Systems Stability and Control”, McGraw-Hill Publishing Company Ltd.
3. T. K. Nagsarkar and M.S. Sukhiza, “Power System Analysis”, Oxford University Press.

Tutorial Component: ***(15 Hours)***

(Note: The course instructor will design tasks to complete the tutorial component of the course.)

Analytical Instrumentation (DSE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Analytical Instrumentation	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To introduce the students to the basics of analytical instrumentation techniques and their applications.
2. To learn quantitative and qualitative analysis techniques of samples of industrial gases and products.
3. To teach the students how to analyze data obtained from analytical instrumentation and interpret the results.
4. To familiarize the students with the principles of analytical instruments like spectrometers, chromatographs, and biosensors.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand knowledge on fundamental principles and concepts of analytical instruments.
2. Describe the various processes for analyzing the samples of industrial products.
3. Apply the knowledge of sampling and analyzing of the industrial products.
4. Evaluate and judge the accuracy of the process for analyzing samples.

UNIT-I ***(13 hours)***

Analytical Spectroscopy Instruments: Elements and types of Analytical Instruments, Ultraviolet and Visible Absorption Spectroscopy, Different types of Spectrophotometers, Sources of Errors and Calibration, Infrared Spectrophotometers – Basic Components and Types, Sample Handling Techniques. Instruments for absorption photometry, single beam and double beam spectrophotometer.

UNIT-II ***(10 hours)***

Flame Emission and Atomic Absorption Spectroscopy: Introduction, Instrumentation for flame spectrometric methods, Flame emission spectrometry, atomic absorption spectrometry, atomic fluorescence spectrometry, Interferences associated with Flames & furnaces, applications, comparison of FES and AAS.

UNIT-III ***(12 hours)***

Chromatography: Gas Chromatograph – Basic Parts of a Gas Chromatograph, Methods of Measurement of Peak Areas, Liquid Chromatograph – Types, High Pressure Liquid Chromatograph**.**
pH meters and Ion Analyzers: Principle of pH Measurement, Electrodes for pH Measurement, pH Meters, Ion Analyzers, Blood pH Measurement.

UNIT-IV ***(10 hours)***

Radiochemical Instruments: Fundamentals of Radiochemical Methods, Radiation Detectors, Liquid Scintillation Counters, Gamma Spectroscopy. X-Ray Spectrometers: Instrumentation for X-Ray Spectrometry, X-Ray Diffractometers, X-Ray Absorption Meters, Electron Probe Micro analyzer.

Essential Readings:

1. Braun, R. D., Introduction to Instrumental Analysis, McGraw Hill, 2008.
2. Khandpur, R. S., Handbook of Biomedical Instrumentation, Tata McGraw Hill, 2000.
3. Currell, G., Analytical Instruments: Performance Characteristics and Quality, Wiley 2008.

Suggested Readings:

1. Willard, H. H., Merritt, L. L., Dean, J. A., Settle, F. A., "Instrumental Methods of Analysis", CBS Publishing & Distribution, 1990.
2. Patranabis, D. Principles of Industrial & Instrumentation, Tata McGraw Hill, 1998.

Practical Component: ***(30 hours)***

List of Experiments:

1. Test and calibrate the spectrophotometer.
2. Measure the percentage transmission, absorption, and concentration of the given sample using a spectrophotometer.
3. Study of each part of the gas chromatograph.
4. Analyze the given gas mixture using a gas chromatograph.
5. Measure refractive index using a refractometer.
6. Analyze the given sample using a refractometer.
7. Measure the pH of the given solution using the double electrode method.
8. Analyze the given sample using a colorimeter.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Biomedical Instrumentation (DSE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Biomedical Instrumentation	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To explore the human body parameter measurements setups.
2. To introduce fundamentals of human physiology and their qualitative measurement.
3. To recognize the therapeutic methods of treatment and the associated instrumentation.
4. To acquire knowledge about the origin of bio-potential, bio-signals and their measurement.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand the problem, identify and formulate solutions in the field of Bio-Medical Engineering for current and future issues.
2. Analyze the cardiac, brain and muscular physiological systems with the related diagnostic measurement methods.
3. Describe the principle and working of cardiac pacemakers, defibrillators, BP measurement, blood flow meters, CO measurement, respiration measurements and their implementation.
4. Analyze various measurement techniques for human physiological parameters.

UNIT-I ***(10 hours)***

Fundamentals of Biomedical Instrumentation: Sources of biomedical signals, Basic Medical Instrumentation system, Performance requirements of medical instrumentation systems. PC based medical instruments, General constraints in design of biomedical instrumentation systems.

Bioelectric Signals and Electrodes: Origin of Bioelectric signals, Types of bioelectric signals-ECG, EEG, EMG, Recording electrodes like Electrode – Tissue interface, polarization, skin contact-impedance, Silver-silver chloride electrodes, Electrodes for ECG, EEG, EMG, Microelectrodes.

UNIT-II ***(13 hours)***

Patient Monitoring System: Bedside patient monitoring systems, Central monitors, Measurement of heart rate – Average heart rate meter, Instantaneous heart rate meter, Measurement of pulse rate, Definition of oximeter & Pulse oximeter.

Blood Pressure Measurement: Introduction, Indirect methods of blood pressure measurement: Korotkoff‟s method, Rheographic method, differential auscultatory technique, Oscillometric technique.

Measurement of Respiration Rate: Impedance pneumography, CO2 method of respiration rate measurement, Apnoea detectors.

UNIT-III ***(10 hours)***

Blood Flow Measurement: Electromagnetic blood flow meter- Principle and Square wave electromagnetic flowmeter. Doppler shift blood flow velocity meter, Blood flow measurement by Doppler imaging.

Cardiac Output Measurement: Measurement of continuous cardiac output derived from the aortic pressure waveform, ultrasound method.

Cardiac Pacemakers and Defibrillators: Need for cardiac pacemaker, External pacemaker, Implantable pacemaker, Types of Implantable pacemakers, Programmable pacemakers, Power sources for Implantable pacemaker. Need for a Defibrillator, DC defibrillator, Pacer-Cardioverter-Defibrillator.

UNIT-IV ***(12 hours)***

Electrocardiograph: Physiology of the heart, Electrical activity of the heart and Electrocardiogram (ECG), Normal & Abnormal cardiac Rhythms, Block diagram-description of an Electrocardiograph, ECG leads, Effects of artifacts on ECG Recordings, Multi- channel ECG machine.

Electroencephalograph: Block diagram description of an Electroencephalograph, 10-20 electrode systems, computerized analysis of EEG. Electromyograph, Biofeedback instrumentation.

Therapeutic Instruments: Cardiac-assist devices, Pump oxygenators, Total artificial heart, Hemodialysis, Lithotripsy, Ventilators, Infant incubators, Drug infusion pumps, Ambulatory and Implantable Infusion systems, Anesthesia Machines, Electrosurgical unit.

Essential Readings:

1. R. S. Khandpur, 'Handbook of Bio-Medical instrumentation', Tata McGraw Hill Publishing Co Ltd., 2003.
2. Leslie Cromwell, Fred J. Weibell, Erich A. Pfeiffer, 'Bio-Medical Instrumentation and Measurements', second edition, Pearson Education, 2002.
3. D. Patranabis, 'Principles of Industrial & Instrumentation', Tata McGraw Hill, 1998.

Suggested Readings:

1. J. Webster, 'Medical Instrumentation', John Wiley & Sons,1995.
2. L. A. Geddes and L. E. Baker, 'Principles of Applied Bio-Medical Instrumentation', John Wiley & Sons, 1975.

Practical Component: ***(30 hours)***

List of Experiments:

1. Perform the measurements based on transducers and plot the characteristics.
2. Compare performance of a variety of electrodes.
3. Measurement of Blood Pressure.
4. Measurement of Blood Glucose level.
5. Analyze Instrumentation amplifier for biomedical signals.
6. Learn pulse oximeter/ Diathermy.
7. Simulate the real time ECG monitoring and ECG wave analysis.
8. Simulate the real time EEG monitoring and EEG wave analysis.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Department of Electrical Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors / Specializations by the Department of Electrical Engineering in Fourth Year

1. **Minor in EE (Offered to ECE and CSE)**
    a. DSE-06/ GE-07: Element of Electrical Power System
    b. DSE-07/ GE-08: Control System Applications in Electrical Engineering
    c. DSE-09/ GE-09: Power Plant Engineering
    d. DSE-09/ GE-09: Machine Learning Applications in Electrical Engineering
    e. DSE-10/ GE-10: Clean Energy Technologies
    f. DSE-10/ GE-10: Introduction to Smart Grid

2. **Minor/Specialization in Robotics and Automation (Offered to EE, ECE, and CSE)**
    a. DSE-06/ GE-07: Nonlinear Control System
    b. DSE-07/ GE-08: Digital Robotic Control
    c. DSE-09/ GE-09: Real Time Embedded System
    d. DSE-09/ GE-09: Process Control
    e. DSE-10/ GE-10: Optimal and Robust Control
    f. DSE-10/ GE-10: Robotics in Biomedical Engineering

3. **Minor/Specialization in Sustainable Energy Engineering (Offered to EE, ECE, and CSE)**
    a. DSE-06/ GE-07: Machine Learning Applications in Sustainable Energy Development
    b. DSE-07/ GE-08: Control of Renewable Energy Integrated Systems
    c. DSE-09/ GE-09: Smart Energy Storage System
    d. DSE-09/ GE-09: Wind and Small Hydro Power Generation
    e. DSE-10/ GE-10: Smart Grid Technologies
    f. DSE-10/ GE-10: Energy Management Systems and SCADA

4. **Minor/Specialization in Electric and Hybrid Vehicle (Offered to EE, ECE, and CSE)**
    a. DSE-06/ GE-07: Electro-Chemistry of Fuel Cells
    b. DSE-07/ GE-08: Modeling and Simulation of EHV
    c. DSE-09/ GE-09: Embedded Based Smart System Design
    d. DSE-09/ GE-09: Computer Aided Design of Electric Machines
    e. DSE-10/ GE-10: EV Charging Infrastructure
    f. DSE-10/ GE-10: Electrical Vehicle System Design

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of Electrical Engineering in SEMESTER VII

Element of Electrical Power System (DSE-06/ GE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Elements of Electrical Power System	4	3	1	0	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To introduce the fundamental structure and components of electrical power systems.
2. To explain the principles of power generation from various sources and basic economic concepts.
3. To analyze transmission line parameters and performance characteristics.
4. To familiarize students with distribution systems and basic power system operation techniques.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Describe the structure of power systems and apply the per-unit system for simple analysis.
2. Explain the working principles of different power generation methods and understand economic operation concepts.
3. Calculate basic transmission line parameters and evaluate line performance using equivalent circuit models.
4. Describe distribution systems and apply introductory concepts of load flow, protection, and faults.

UNIT-I ***(10 hours)***

Introduction to Electrical Power Systems: Structure of power systems including generation, transmission, distribution, and utilization; Load characteristics: types of loads, load curves, load duration curves, diversity factor, load factor; Per-unit system: definition, advantages, base values, per-unit representation of system components.

UNIT-II ***(08 hours)***

Power Generation: Conventional sources: thermal, hydroelectric, and nuclear power plants (working principles, efficiency, and site selection); Non-conventional sources: solar, wind, and biomass (basics of energy conversion and grid integration).

UNIT-III ***(15 hours)***

Power Transmission: Overhead transmission lines: types of conductors, bundle conductors, resistance calculation, skin effect; Inductance and capacitance: calculation for single and three-phase lines, GMD and GMR concepts; Equivalent circuit model: nominal pi and T representations for short, medium, and long lines; Performance analysis: voltage regulation, efficiency, surge impedance loading; Corona: critical disruptive voltage, power loss; HVDC transmission: monopolar and bipolar links, advantages over AC, basic converter operation (rectifier and inverter modes).

UNIT-IV ***(12 hours)***

Power Distribution and Basic Operation: Distribution systems: radial, ring main, and interconnected types; Substation equipment: transformers, circuit breakers, insulators; Voltage drop calculations, power factor improvement using capacitors; Load flow studies: bus classification (PQ, PV, slack), basic concepts and importance; Introduction to symmetrical faults: types and basic concepts; System protection: principles of overcurrent relays, differential relays, distance relays.

Essential Readings:

1. John J. Grainger and William D. Stevenson, Power System Analysis, McGraw Hill Education, 4th Edition, 2017.
2. I. J. Nagrath and D.P. Kothari, Modern Power System Analysis, Tata McGraw Hill, 5th Edition, 2011.
3. Hadi Saadat, Power System Analysis, PSA Publishing, 3rd Edition, 2011.

Suggested Readings:

1. Ashfaq Husain, Electrical Power Systems, CBS, 5th edition, 2018.
2. C. L. Wadhwa, Electrical Power Systems, New Age International Publishers, 8th Edition, 2017.

Tutorial Component: ***(15 hours)***

List of Tutorial Tasks:

1. Problems related to comparison of conventional vs. smart grid features and benefits.
2. Problems related to calculation of drivers and challenges in smart grid adoption.
3. Problems related to identification of smart grid functions from real-world scenarios.
4. Problems related to conceptual modelling of smart grid domains and components.
5. Problems related to per-unit representation and sizing of AMI and PMU in sample systems.
6. Problems related to integration of DER and energy storage in smart grid architecture.
7. Problems related to selection of communication technologies for different smart grid layers.
8. Problems related to interpretation of IEC 61850 standards in substation automation.
9. Problems related to basic cyber-security risk assessment in smart grid communication.
10. Problems related to renewable integration and microgrid operation modes.
11. Problems related to demand response and V2G calculations in smart grid applications.
12. Problems related to economic and policy analysis of smart grid initiatives.

(Note: The course instructor may add/delete/update new tutorial tasks in addition to the above suggested tutorial tasks.)

Control System Applications in Electrical Engineering (DSE-07/ GE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Control System Applications in Electrical Engineering	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To introduce the basic principles of feedback control systems and their modeling.
2. To explain time-domain analysis and stability concepts for control systems.
3. To familiarize students with frequency-domain techniques and controller design.
4. To emphasize practical applications of control systems in electrical engineering domains such as power systems and electric drives.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Model simple electrical systems and analyze their dynamic response using transfer functions.
2. Evaluate stability and performance of feedback control systems in the time domain.
3. Apply frequency response methods for system analysis and basic controller tuning.
4. Understand and describe control applications in electrical power systems and motor drives.

UNIT-I ***(12 hours)***

Introduction to Control Systems: Open-loop and closed-loop control systems, feedback concepts; Mathematical modeling of electrical systems: transfer functions of Electrical circuits; Block diagram representation and reduction, signal flow graphs; Standard test signals, time response specifications.

UNIT-II ***(10 hours)***

Time-Domain Analysis: First and second-order systems, steady-state error, error constants; Stability analysis: Routh-Hurwitz criterion; Root locus technique: basic rules, plotting, and applications to control design.

UNIT-III ***(11 hours)***

Frequency-Domain Analysis: Bode plots, gain margin, phase margin, Nyquist stability criterion; PID controllers: structure, effects of proportional, integral, and derivative actions, basic tuning methods (Zeigler-Nichols).

UNIT-IV ***(12 hours)***

Applications in Electrical Engineering: Speed control of DC motors using armature and field control; Voltage and frequency control in power systems (AVR basics); Load frequency control in power systems; Introduction to control in induction motor drives (V/f control); PLC basics for industrial electrical control.

Essential Readings:

1. Norman S. Nise, Control Systems Engineering, Wiley, 8th Edition, 2020.
2. Katsuhiko Ogata, Modern Control Engineering, Prentice Hall, 5th Edition, 2010.
3. Benjamin C. Kuo, Automatic Control Systems, Prentice Hall, 10th Edition, 2010.

Suggested Readings:

1. I. J. Nagrath and M. Gopal, Control Systems Engineering, New Age International Publishers, 6th Edition, 2018.
2. Gopal K. Dubey, Fundamentals of Electrical Drives, Narosa Publishing House, 2nd Edition, 2001.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To derive the transfer function of a DC motor; plot step response using MATLAB/Simulink.
2. To build simple RLC on breadboard (resistors/capacitors/inductors); measure response with oscilloscope, then simulate in Simulink for comparison.
3. To simulate first/second-order systems in MATLAB; compute error constants for ramp input.
4. To test stability of sample EE systems (e.g., feedback loop); plot poles using MATLAB/Simulink.
5. Frequency response for a PID-controlled system; analyze gain/phase margins.
6. To sketch locus in MATLAB for motor speed control; verify gain effects on a basic op-amp hardware setup.
7. To Implement Zeigler-Nichols on Arduino kit with DC motor (e.g., L298N driver); tune for setpoint tracking.
8. To program simple motor start/stop sequence using open source PLC Simulator (e.g., LogixPro).

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Nonlinear Control System (DSE-06/ GE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Nonlinear Control System	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To understand the fundamentals of non-linear systems.
2. To understand the modeling concepts of non-linear systems.
3. To understand the stability of non-linear systems.
4. To acquire the knowledge of various non-linear control techniques.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Model the non-linear systems considering practical applications.
2. Design and analyze the non-linear control systems.
3. Analyze the stability of non-linear systems.
4. Apply practically the various concepts on the non-linear systems under different conditions.

UNIT-I ***(15 hours)***

Introduction: Introduction to non-linear system, General Properties of linear and non-linear systems, perturbation theory, and perturbation dynamics, controllability and observability of non-linear systems, Lipschitz continuity, existence and uniqueness of solution of non-linear systems, common differences with linear system, types of nonlinearities, inherent nonlinearity (Saturation, dead zone, Hysteresis, Back-lash), intentional non-linearity, non-linear phenomenon (Frequency-amplitude dependence, multi-valued response and jump resonance, sub-harmonic oscillation, limit cycle, Frequency entrainment, Asynchronous quenching).

UNIT-II ***(10 hours)***

Modeling of Non-Linear Systems: Modeling of simple mechanical systems, degree-of freedom, configuration spaces and state-space representation, equilibrium points/operating points, Jacobian linearization.

Mathematical Notions: Notion of vector field, trajectories, vector field plot, phase plane portrait, positively invariant sets and classification of equilibrium points.

UNIT-III ***(12 hours)***

Qualitative Analysis of Second Order Systems: Second-order systems, periodic solution, Bendixson's theorem and Poincare-Bendixson criteria, phase plane analysis: Introduction to phase-plane analysis, method of Isoclines for constructing Trajectories, singular points, phase-plane analysis of nonlinear control systems.

Notions of Stability of Non-Linear Systems: Lyapunov theorem, small gain theorem, describing function method, asymptotic stability, exponential stability.

UNIT-IV ***(08 hours)***

Lyapunov's Stability Notions: Lyapunov's direct and indirect stability, La-Salles's invariance principle and its examples.
Non-Linear Control Techniques: Feedback linearization, backstepping, variable structure control, nonlinear observers.

Essential Readings:

1. Nonlinear Systems, H Khalil, Macmillan, 3rd edition, 2002.
2. Nonlinear systems: analysis, stability, and control, Springer Science & Business Media, Shankar Sastry, 2013 .
3. Non-linear differential equations, Elsevier by Sansone, Giovanni, and Roberto Conti., 2014 .

Suggested Readings:

1. Nonlinear Control Systems: Analysis and Design, H J Marquez, Hoboken: Wiley-Interscience, 2003.
2. Applied Nonlinear control by J J Slotine and W P Li, Prentice Hall, 1991.
3. Analysis and Design of Nonlinear Control Systems by A Astolfi, L Marconi, 2008.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To study saturation and dead zone non-linearity using describing function technique of a relay control system
2. To draw phase trajectory of a given non-linear system.
3. To study and analysis of Pendulum/inverted pendulum system.
4. To study and analysis of Mass-Spring system.
5. To study and analyze Limit cycles and Phase portraits for nonlinear systems.
6. To study the existence of periodic orbits in different nonlinear systems.
7. To study and perform bifurcation Analysis of nonlinear system.
8. Simulations on the describing function method for other common nonlinearities.
9. To design a controller for a nonlinear system (SMC, Adaptive, Backstepping, NN Based, Feedback linearization).
10. Case Study-1
11. Case Study-2

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Digital Robotic Control (DSE-07/ GE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Robotic Control	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To understand the basic terminologies and concepts associated with Robotics and end effectors.
2. To analyze and understand the transformation of the end effectors' position and be familiar with the kinematics and dynamics of robots.
3. To understand the robot work cell layout and the recent trends in the application of artificial intelligence and expert systems in robotics.
4. To acquire knowledge on emerging developments in robots.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Recognize the basics of robots and apply the concepts associated with end effectors.
2. Acquire the knowledge about robotic control system, feedback devices and sensors.
3. Evaluate transformation of end effectors position and being familiar with the kinematics and dynamics of robots.
4. Analyze the robot work cell layout and application of artificial intelligence in robotics.

UNIT-I ***(15 hours)***

Introduction to Robotics and End Effectors: Introduction to Robotic, History - Developments in Robotics, Robot anatomy, Definition and law of robotics, Terminology of Robotics, Accuracy and repeatability, dexterity, compliance, RCC device of Robotics-Simple problems, Specifications of Robot-Speed of Robot- Robot joints and links, Robot Drive systems-Hydraulic, Pneumatic and Electric system. End Effectors - Mechanical grippers-Slider crank mechanism, cam type, Screw type, Rotary actuators, Magnetic grippers, Vacuum grippers, Gripper force analysis, Gripper selection and design considerations.

UNIT-II ***(12 hours)***

Robot Controls and Sensors: Control system for robot joint- Controllers, Control system analysis – Transient Response, Steady state, Trajectory planning of end effectors. Feedback devices-Encoder, Resolver, LVDT-Motion Interpolations, Adaptive control Sensors in robot, Characteristics of sensing devices, Selections and need of sensors, Touch sensors-Tactile sensor, Proximity and range sensors, Force sensor-Light sensors, Pressure sensors, Robotic vision sensor.

UNIT-III ***(10 hours)***

Robot Transformations, Kinematics and Dynamics Robot kinematics-Types-2D, 3D Transformation, inverse kinematics, Scaling, Rotation, Translation, Homogeneous Transformations, multiple transformations. Kinematic equations using Homogeneous Transformations - Joints, frame assignment to links, Orientation, direct kinematics Solving Kinematic equations – Velocities and Static forces in manipulators:- Jacobians, singularities, static forces, Jacobian in force domain, Dynamics:- Introduction to Dynamics.

UNIT-IV ***(08 hours)***

Robot Cell Design and Artificial Intelligence Applications Robot work cell design - control-Sequence control operator interface, safety monitoring devices in Robot Mobile robot working principle, actuation using MATLAB, NXT Software Introduction to Artificial Intelligence, Need and application of AI techniques, Artificial neural networks in manufacturing automation, Fuzzy decision and control, robots and application of robots using AI and fuzzy.

Essential Readings:

1. Robotics Engineering an Integrated Approach, Phi Learning., Richard D. Klafter, Thomas. A, Chri Elewski, Michael Negin, 2009.
2. Engineering Foundation of Robotics, Prentice Hall Inc., Francis N. Nagy, Andras Siegler, 1987.
3. Digital control system analysis and design. Prentice Hall Press Phillips, Charles L., and H. Troy Nagle. 2007.

Suggested Readings:

1. Robotics Technology and flexible automation,Tata McGraw-Hill Education, Deb. S.R, 2009.
2. Technology Programming and Applications, McGraw Hill, Mikell P Groover & Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, Industrial Robotics, 2012.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Introduction to programming using Arduino ide, Programming inputs and outputs.
2. Interfacing Arduino with LED to Blink and fade.
3. Interfacing Arduino with Buzzer.
4. Interfacing digital read - proximity sensor and serial monitor with Arduino.
5. Interfacing digital input - ultrasonic sensor with Arduino for measuring distance.
6. Bidirectional speed control of DC motor through L293D or ULN2003 interface.
7. Interfacing analog output -Servo Motor with Arduino.
8. Angular positioning of servo motor by using the ultrasonic sensor with Arduino.
9. Controlling servo motor by using the Potentiometer with Arduino.
10. Interfacing Bluetooth Module and IR Sensor with Arduino.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Machine Learning Applications in Sustainable Energy Development (DSE-06/ GE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Machine Learning Applications in Sustainable Energy Development	4	3	0	1	Fundamentals of Computer Programming

Course Objectives:

1. To introduce fundamental concepts of artificial intelligence and machine learning relevant to energy systems.
2. To develop computational proficiency in Python for data analysis and ML implementation.
3. To provide a strong understanding of supervised and unsupervised learning techniques.
4. To enable students to formulate and solve optimization problems encountered in renewable energy systems.
5. To apply machine learning methods to real-world problems in alternative energy generation, forecasting, and system diagnostics.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain core concepts of machine learning and implement basic ML workflows using Python.
2. Apply supervised learning techniques for regression and classification problems in energy datasets.
3. Analyze unlabeled energy data using clustering and probabilistic learning methods.
4. Formulate and solve optimization problems relevant to renewable energy systems.
5. Design ML-based solutions for forecasting, energy estimation, demand analysis, and fault detection in alternative energy applications.

UNIT-I ***(15 hours)***

Foundations of Machine Learning and Python Programming: Introduction to artificial intelligence and machine learning; types of machine learning; ML workflow; data representation and preprocessing. Basics of Python programming for ML: variables, data types, control structures, functions, libraries for numerical computing and data handling.

UNIT-II ***(10 hours)***

Supervised Learning Techniques: Regression methods: linear regression, gradient descent optimization, weighted least squares, and logistic regression. Classification methods: k-Nearest Neighbors (kNN), discriminant function analysis, Bayesian decision theory, and decision tree learning.

UNIT-III ***(10 hours)***

Unsupervised Learning and Optimization Techniques: Clustering techniques: k-means clustering, hierarchical clustering, Optimization fundamentals: problem formulation, decision variables, objective functions, constraints, maxima and minima; solving optimization problems relevant to engineering applications.

UNIT-IV ***(10 hours)***

Machine Learning Applications in Alternate Energy Systems: Weather and solar/wind data forecasting; estimation of energy generation from renewable energy sources; building energy demand prediction; system fault detection, diagnostics, and performance monitoring using ML techniques.

Essential Readings:

1. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 1st Edition, 2012.
2. B. K. Bose, *Artificial Intelligence Techniques in Smart Grid and Renewable Energy Systems*, Springer, 1st Edition, 2018.
3. E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 3rd Edition, 2014.

Suggested Readings:

1. R. Whig, D. Sharma, S. Aneja, A. Elngar, and J. Silva, *Artificial Intelligence and Machine Learning for Sustainable Development: Innovations, Challenges, and Applications*. Boca Raton, FL, USA: CRC Press, 2024.
2. A. C. Müller and S. Guido, *Introduction to Machine Learning with Python*, O'Reilly Media, 1st Edition, 2016.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To study Python programming basics for data analysis using NumPy, Pandas, and Matplotlib.
2. To study data preprocessing and feature extraction techniques for renewable energy datasets using Python.
3. To study the implementation of linear regression for solar and wind power generation prediction using MATLAB/Python.
4. To study gradient descent–based optimization for training regression models using MATLAB/Python.
5. To study logistic regression for the classification of operating conditions in energy systems using MATLAB/Python.
6. To study k-Nearest Neighbor (kNN)–based classification for fault identification in energy data using MATLAB/Python.
7. To study decision tree–based classification for system condition monitoring using Python.
8. To study formulation and solution of optimization problems for improving energy system performance using MATLAB/Python.
9. To study weather data–based forecasting of renewable energy generation using machine learning models using Python.
10. To study machine learning–based fault detection and diagnostics in renewable energy systems using Python.

(Note: The Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Control of Renewable Energy Integrated Systems (DSE-07/ GE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Control of Renewable Energy Integrated Systems	4	3	0	1	Energy and Its Resources

Course Objectives:

1. To provide a fundamental understanding of photovoltaic and wind energy conversion systems.
2. To develop knowledge of modelling techniques for PV and wind energy systems.
3. To introduce control strategies for standalone, grid-connected, and hybrid renewable energy systems.
4. To enable students to simulate and analyze renewable energy systems and their control mechanisms.
5. To expose students to recent research trends and intelligent control applications in renewable energy systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the principles, classification, benefits, and challenges of renewable energy integrated systems.
2. Model photovoltaic cells, PV arrays, and wind energy systems using mathematical and simulation tools.
3. Develop and simulate control strategies for standalone and grid-connected renewable energy systems.
4. Evaluate system performance using case studies and interpret recent research developments in intelligent control of renewable energy systems.

UNIT-I ***(06 hours)***

Introduction to Renewable Energy Systems: Overview of renewable energy sources: photovoltaic and wind energy systems. Classification of renewable energy integrated systems. Benefits and challenges of renewable energy systems. Solar energy potential and assessment.

UNIT-II ***(10 hours)***

Photovoltaic and Wind Energy Systems: Modelling of photovoltaic cells and PV arrays. Maximum Power Point Tracking (MPPT): concept and need, MPPT algorithms and methods. Wind energy fundamentals: wind speed and power relationship (Betz's Law), wind speed distribution, Weibull probability function. Types of wind turbines. Wind turbine power curve and performance metrics.

UNIT-III ***(14 hours)***

Modelling and Control of Renewable Energy Systems: Mathematical modelling of PV systems and wind turbine systems. MPPT techniques: Perturb & Observe (P&O), Incremental Conductance (I&C). Control of renewable energy systems: grid-connected and standalone systems. Introduction to conventional control mechanisms used in renewable energy systems.

UNIT-IV ***(15 hours)***

Control Applications and Case Studies: Design and control of grid-connected, standalone, and grid-interactive renewable energy systems. Case studies related to PV, wind, and hybrid systems. Recent research and developments in intelligent control techniques for renewable energy systems and their applications.

Essential Readings:

1. J. A. Duffie and W. A. Beckman, *Solar Engineering of Thermal Processes*, 4th ed. Hoboken, NJ, USA: Wiley, 2013.
2. J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind Energy Explained: Theory, Design and Application*, 2nd ed. Chichester, U.K.: Wiley, 2009.
3. M. Rahman and Y. Khalid, Eds., *Artificial Intelligence in Renewable Energetic Systems: Smart Technologies for Sustainable Development*. Boca Raton, FL, USA: CRC Press, 2022.

Suggested Readings:

1. S. Chakraborty, M. G. Simões, and W. E. Kramer, Eds., *Power Electronics for Renewable and Distributed Energy Systems*. New York, NY, USA: Springer, 2013.
2. J. Momoh, *Smart Grid: Fundamentals of Design and Analysis*. Hoboken, NJ, USA: Wiley-*IEEE Press*, 2012.
3. Journal Research papers from publishers like IEEE/Elseviers/Springers etc.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Modelling and simulation of a photovoltaic cell and PV array characteristics.
2. Implementation and performance analysis of P&O MPPT algorithm for a PV system.
3. Implementation and comparison of the Incremental Conductance MPPT technique for PV systems.
4. Modelling of wind turbine and analysis of wind speed–power characteristics.
5. Simulation of wind turbine power curve and performance metrics.
6. Modelling and control of a standalone photovoltaic system with DC–DC converter.
7. Modelling and simulation of a grid-connected PV system with inverter control.
8. Modelling and simulation of a standalone wind energy conversion system.
9. Performance analysis of a hybrid PV–wind renewable energy system.
10. Case study–based simulation of a grid-interactive renewable energy system with basic control strategies.
11. SIL/HIL implementation of PV system model using a digital real-time simulator.
12. SIL/HIL modelling of wind energy conversion system and analysis of dynamic response under wind variations.
13. SIL/HIL-based control and performance evaluation of a hybrid PV–wind system under grid-connected and standalone modes.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Electro-Chemistry of Fuel Cells (DSE-06/ GE-07)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electro-Chemistry of Fuel Cells	4	3	0	1	Introduction to Electrical and Electronics Engineering, Physics

Course Objectives:

1. Understanding of electrochemical principles and introducing fuel cells in the EV context
2. Dive into reaction mechanisms, kinetics, and performance-limiting factors in fuel cells.
3. Cover key components and transport processes critical for EV fuel cell stacks
4. Apply concepts to real-world EV integration, performance analysis, and future outlook

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the working principles, thermodynamics, and electrochemical processes of various fuel cell types.
2. Analyze and compare the performance, advantages, and limitations of different fuel cells such as PEMFC, SOFC, AFC, and others.
3. Evaluate the impact of key fuel cell components and fueling technologies, including hydrogen storage and reforming methods, on system performance.
4. Apply knowledge of fuel cell technologies to practical applications like power generation, transportation systems, and hybrid electric drivetrains while assessing their economic and environmental implications.

UNIT-I ***(12 hours)***

Fundamentals of Electrochemistry and Fuel Cells:

Introduction to electrochemistry: Galvanic vs. electrolytic cells, electrode potentials, Nernst equation, standard hydrogen electrode, reference electrodes. Thermodynamics of electrochemical reactions: Gibbs free energy, cell potential (E°), reversible work, efficiency (theoretical vs. practical), heat of reaction, entropy changes. Fuel cells overview: Definition, difference from batteries (continuous fuel supply), advantages over IC engines (higher efficiency ~50–60%, zero tailpipe emissions), role in EVs (FCEVs as range-extended or pure hydrogen vehicles). Classification of fuel cells: Focus on low-temperature types relevant to EVs (PEMFC, AFC); brief on others (SOFC, MCFC, PAFC, DMFC). Basic fuels for fuel cells: Hydrogen (primary for EVs), challenges with impurities (CO, S poisoning), alternative fuels (methanol, hydrocarbons). Why fuel cells in EVs: Comparison with battery EVs (BEVs) — refueling time, range, energy density.

UNIT-II ***(12 hours)***

Electrochemistry of Fuel Cells – Kinetics and Losses

Electrode kinetics: Butler-Volmer equation, exchange current density, Tafel equation, activation overpotential (η_act). Types of voltage losses/polarizations: Activation, ohmic, concentration (mass transport) losses; polarization curve (V-I characteristics). Fuel cell efficiency: Voltage efficiency, fuel utilization, overall efficiency calculation. Half-cell reactions in PEM fuel cells: Electrocatalysts: Platinum (Pt) and Pt-based alloys, issues with Pt loading/cost, carbon supports, degradation mechanisms (Pt dissolution, agglomeration). ORR kinetics challenges: Slow 4-electron transfer, peroxide formation, poisoning effects.

UNIT-III ***(11 hours)***

Components, Transport Phenomena, and PEM Fuel Cell Design

Proton exchange membrane (PEM): Nafion structure, proton conductivity, water management (humidification needs), thickness effects. Membrane electrode assembly (MEA): Catalyst layers, gas diffusion layers (GDL), microporous layers. Bipolar plates: Materials (graphite, metallic), flow field designs (parallel, serpentine), contact resistance. Mass transport and charge transport: Gas diffusion, proton migration, electron conduction, water transport (electro-osmotic drag, back-diffusion). Heat and water management: Thermal balance, cooling systems, flooding/dehydration issues in PEMFCs. Fuel cell stack for EVs: Series/parallel configuration, power output scaling, system auxiliaries (air compressor, hydrogen recirculation, humidifiers).

UNIT-IV ***(10 hours)***

Performance, Modeling, Applications, and Challenges in EVs

Fuel cell performance testing: Polarization curves, power density, efficiency maps, durability testing.

Modeling of fuel cells: Simple 0D models (equivalent circuit), basic equations for voltage losses, introduction to CFD/1D models for transport. Fuel cell electric vehicles (FCEVs): System integration (fuel cell + battery hybrid), powertrain architecture, hydrogen storage (compressed, liquid), refueling infrastructure. Case studies: Toyota Mirai, Hyundai Nexo — stack specs, efficiency, range. Challenges and advancements: Cost reduction (low-Pt catalysts), durability (>5000–10000 hours for automotive), cold-start issues, hydrogen production/storage/safety. Future in EVs: Role in heavy-duty/long-haul vehicles, comparison with BEVs, policy support (FAME schemes, green hydrogen in India).

Essential Readings:

1. EG&G Technical Services, Inc., *Fuel Cell Handbook*, 7th Edition, U.S. Department of Energy, Office of Fossil Energy, 2004.
2. Mehrdad Ehsani, Yiming Gao, Stefano Longo, and Kambiz Ebrahimi, *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*, 3rd Edition, CRC Press, 2018.
3. B. Viswanathan and M. Aulice Scibioh, *Fuel Cells: Principles and Applications*, Universities Press.
4. Gregor Hoogers, *Fuel Cell Technology Handbook*, CRC Press / SAE International, 2003.

Suggested Readings:

1. B. H. Khan, Non-Conventional Energy Resources, 2nd Edition, Tata McGraw-Hill, 2013.
2. B. Hart and G. J. Womack, Fuel Cells: Theory and Applications, Chapman and Hall.
3. Fuel Cells for Automotive Applications, Professional Engineering Publishing, UK, 2004.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To demonstrate the electrolysis of water using a reversible PEM fuel cell and verify the stoichiometry of hydrogen and oxygen production.
2. To assemble a single PEM fuel cell and measure its open-circuit voltage under standard operating conditions.
3. To obtain the polarization (V-I) curve and power density curve of a PEM fuel cell and identify different regions of voltage losses.
4. To study the effect of operating temperature on the performance characteristics of a PEM fuel cell.

5. To investigate the influence of gas humidification (relative humidity) on the voltage and performance of a PEM fuel cell.
6. To determine the fuel utilization and overall efficiency of a PEM fuel cell under constant load conditions.
7. To construct a Tafel plot from low-current data and analyze the activation overpotential for the oxygen reduction reaction (ORR).
8. To demonstrate cathode flooding and its impact on mass transport limitations in a PEM fuel cell at high current densities.
9. To evaluate the effect of reactant gas stoichiometry (flow rates) on the polarization behavior and concentration polarization of a PEM fuel cell.
10. To demonstrate the basic operation of a fuel cell + battery hybrid system simulating an EV-like dynamic load profile and observe power sharing.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Modeling and Simulation of EHV (DSE-07/ GE-08)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Modeling and Simulation of EHV	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To understand the evolution, architecture, and interdisciplinary nature of EVs and HEVs.
2. To develop mathematical and physical models of EV/HEV subsystems.
3. To apply simulation tools, Simulink/Simscape, for system-level analysis.
4. To study control strategies for electric drives and hybrid vehicle architectures.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain EV and HEV architectures and their operating principles.
2. Develop mathematical models for vehicle dynamics and energy storage systems.
3. Model and simulate electric motors and power electronic converters.
4. Analyze AC motor control techniques for traction applications.
5. Simulate and evaluate complete EV/HEV systems using software tools.

UNIT-I ***(09 hours)***

Introduction, Architectures, and Modeling: Introduction to EVs and HEVs, brief history of electric and hybrid vehicles, comparison of EVs and HEVs, interdisciplinary nature of hybrid electric vehicles. series hybrid architecture, parallel hybrid architecture, split power (series–parallel) architecture, system modeling fundamentals, mathematical modeling techniques, overview of software tools for EV/HEV modeling.

UNIT-II ***(12 hours)***

Simscape and Vehicle Dynamics: Physical network approach to modeling: variable types, direction of variables, passive and active elements, physical conserving ports, physical signal ports, getting started with Simscape, longitudinal vehicle load modeling, tractive force calculation and road load modeling.

Energy Storage Modeling: Battery modeling fundamentals, charge and discharge dynamics, State of Charge estimation, thermal modeling of batteries, ultracapacitor modeling.

UNIT-III ***(14 hours)***

DC Motor Modeling and Control: Permanent magnet DC motor modeling, speed and torque control of DC motors, DC motors as traction motors, separately excited DC motor modelling, four-quadrant operation.

Power Electronics and Hardware Control: Role of power electronics in EVs, AC–DC conversion, DC–DC converters: buck and boost, DC motor control using power electronic converters, DC–AC conversion: three-phase inverter, PWM control of inverters.

UNIT-IV ***(10 hours)***

AC Motor Modeling for Traction: Introduction to AC motors for EVs, d–q modeling and vector control, PMSM modeling and control, AC motor drive simulation for traction applications.

Hybrid Vehicle System Modeling: Modeling of series hybrid electric vehicle, modeling of parallel hybrid electric vehicle, split power HEV modeling, overall vehicle supervisory control, performance analysis under different drive cycles.

Essential Readings:

1. Iqbal Husain, Electric and Hybrid Vehicles: Design Fundamentals, CRC Press.
2. James Larminie & John Lowry, Electric Vehicle Technology Explained, Wiley.
3. Mehrdad Ehsani et al., Modern Electric, Hybrid Electric, and Fuel Cell Vehicles, CRC Press.

Suggested Readings:

1. Shuvra Das, Modeling for Hybrid and Electric Vehicles Using Simscape, Elsevier.
2. MATLAB & Simulink Documentation – MathWorks.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Modeling of vehicle longitudinal dynamics in Simulink.
2. Simulation of vehicle road load and tractive effort.
3. Battery modeling with SOC and thermal effects.
4. Ultracapacitor modeling and comparison with battery.
5. PMDC motor modeling and speed control.
6. Four-quadrant DC motor control using DC–DC converter.
7. Modeling of buck and boost converters for EV applications.
8. PMSM modeling with vector (d–q) control.
9. Simulation of a series hybrid electric vehicle.
10. Simulation of a parallel or split-power HEV under a standard drive cycle.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of Electrical Engineering in SEMESTER VIII

Power Plant Engineering (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power Plant Engineering	4	3	1	0	Introduction to Electrical and Electronics Engineering, Mathematics-I, Physics

Course Objectives:

1. To introduce sources of energy and the layout of various power plants.
2. To explain the working and performance of steam power plants.
3. To describe the operation of hydroelectric, diesel, and gas turbine power plants.
4. To provide basics of nuclear power plants and power plant economics.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Identify sources of energy and describe general layouts of power plants.
2. Analyze steam cycles and compute performance parameters of thermal plants.
3. Explain components and operation of hydroelectric, diesel, and gas turbine plants.
4. Understand nuclear reactor types and basic economic aspects of power generation.

UNIT-I ***(12 hours)***

Introduction: Sources of energy, types of power plants, review of thermodynamic cycles in power plants; Fuels and combustion: Coal handling, pulverizers, ash handling, draught systems; Steam Power Plant: General layout, Rankine cycle, reheat and regenerative cycles, efficiency improvement.

UNIT-II ***(10 hours)***

Steam Generators and Turbines: Boilers: fire tube and water tube, high pressure boilers; Steam nozzles and turbines: impulse and reaction types, velocity diagrams, compounding; Condensers and cooling towers.

UNIT-III ***(13 hours)***

Hydroelectric and Diesel/Gas Turbine Plants: Hydroelectric plants: classification, components (penstock, surge tank), turbines (Pelton, Francis, Kaplan); Diesel engine power plants: layout, supercharging; Gas turbine plants: open and closed cycles, components, intercooling, reheating, regeneration.

UNIT-IV ***(10 hours)***

Nuclear Power Plants and Economics: Nuclear fission, reactor types (PWR, BWR), components, safety features; Power plant economics: load curves, load factor, capacity factor, cost analysis, tariffs.

Essential Readings:

1. P.K. Nag, Power Plant Engineering, Tata McGraw Hill, 4th Edition, 2014.
2. R.K. Rajput, Power Plant Engineering, Laxmi Publications, 5th Edition, 2016.

3. Arora and Domkundwar, A Course in Power Plant Engineering, Dhanpat Rai & Co., Latest Edition.

Suggested Readings:

1. G.R. Nagpal and S.C. Sharma, Power Plant Engineering, Khanna Publishers, Latest Edition.
2. Frederick T. Morse, Power Plant Engineering, East-West Press, Latest Edition.

Tutorial Component: ***(15 hours)***

List of Tutorial Tasks:

1. Problems related to classification of power plants based on energy sources and site selection criteria.
2. Problems related to efficiency calculations for Rankine, reheat, and regenerative steam cycles.
3. Problems related to coal handling systems, pulverization, and draught system design considerations.
4. Problems related to classification and performance analysis of different types of boilers (fire-tube vs. water-tube).
5. Problems related to velocity diagrams, compounding methods, and efficiency of impulse vs. reaction turbines.
6. Problems related to condenser performance, vacuum efficiency, and cooling tower heat balance calculations.
7. Problems related to classification of hydroelectric plants and selection of suitable turbines (Pelton, Francis, Kaplan).
8. Problems related to layout design and supercharging effects in diesel engine power plants.
9. Problems related to open/closed cycle analysis of gas turbine plants with intercooling, reheating, and regeneration.
10. Problems related to nuclear reactor types, fuel-moderator-coolant combinations, and safety feature comparisons.
11. Problems related to load curves, load factor, capacity factor, and plant use factor calculations.
12. Problems related to economic analysis including depreciation methods, tariff calculation, and cost comparison of different power plants.

(Note: The course instructor may add/delete/update new tutorial tasks in addition to the above suggested tutorial tasks.)

Machine Learning Applications in Electrical Engineering (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Machine Learning Applications in Electrical Engineering	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I, Fundamentals of Computer Programming

Course Objectives:

1. To introduce the fundamentals of machine learning and its relevance to engineering problems.
2. To explain supervised and unsupervised learning techniques with a focus on regression and classification.
3. To cover optimization methods and model evaluation for machine learning models.
4. To emphasize applications of machine learning in electrical engineering domains such as power systems and energy management.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand basic concepts of data science and machine learning, including supervised and unsupervised methods.
2. Apply regression and classification algorithms to solve simple engineering problems.
3. Implement optimization and evaluation techniques using Python tools.
4. Analyze machine learning applications in electrical engineering, such as load forecasting and fault detection.

UNIT-I ***(11 hours)***

Introduction to Machine Learning: Differences between supervised, unsupervised, and reinforcement learning; Probability and statistics basics (sample/population properties, covariance, correlation); Motivation for ML in electrical engineering (e.g., predictive maintenance in power grids).

UNIT-II ***(11 hours)***

Supervised Learning - Regression and Classification: Linear regression, model fitting, coefficient of determination, confidence intervals; Overfitting/underfitting, bias-variance tradeoff, regularization (ridge, LASSO); Logistic and softmax regression; Applications in EE (e.g., load forecasting using regression).

UNIT-III ***(12 hours)***

Unsupervised Learning and Advanced Techniques: Clustering algorithms (k-means, density-based); Dimensionality reduction (SVD, PCA, nonlinear methods like UMAP, t-SNE); Decision trees, ensemble methods (bagging, boosting, random forests); Support vector machines for classification/regression; Hyperparameter tuning (grid search, cross-validation).

UNIT-IV ***(11 hours)***

Neural Networks and EE Applications: Introduction to neural networks (feed-forward, convolutional, recurrent); Loss functions, gradient descent variants (SGD, Adam); Applications in electrical engineering: Fault detection in transmission lines, renewable energy prediction (solar/wind), smart grid optimization, energy consumption forecasting.

Essential Readings:

1. Raghunathan Rengaswamy and Resmi Suresh, Data Science for Engineers, CRC Press, 2022.
2. Ethem Alpaydin, Introduction to Machine Learning, MIT Press, 4th Edition, 2020.
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.

Suggested Readings:

1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 3rd Edition, 2022.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Basic Statistics Visualization: Compute sample/population mean, covariance, and correlation on electrical load datasets; plot results to analyze grid stability relationships.
2. Fit linear regression on historical power demand data, evaluate R-squared and confidence intervals for EE predictive maintenance.
3. Train ridge/LASSO models on noisy transmission line fault data; compare overfitting via bias-variance plots.
4. K-Means Clustering: Cluster renewable energy output (solar/wind) data into demand patterns; visualize with EE scatter plots.
5. PCA Dimensionality Reduction: Apply PCA/SVD on high-dimensional smart grid sensor data; compare variance retention for anomaly detection.
6. Build decision trees/random forests for fault classification in power systems; tune via grid search/cross-validation.
7. Use support vector regression to predict energy consumption; test on transmission line datasets.
8. Train feed-forward NN with Adam optimizer for solar power forecasting; evaluate loss on EE time-series data.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Real Time Embedded System (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Real Time Embedded Systems	4	3	0	1	Fundamentals of Computer Programming

Course Objectives:

1. To understand the concept of embedded system design and analysis.
2. To expose the basic concepts of embedded programming.
3. To learn real time operating systems.
4. To acquire knowledge about real time processors and operating systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Apply the concepts of embedded systems and its features.
2. Analyze various Real Time Operating system is used in Embedded System.
3. Design the flow &Techniques to develop Software for embedded system networks.
4. Analyze Real-time applications using embedded System Products.

UNIT-I ***(12 hours)***

Introduction to Embedded System Design: Complex systems and microprocessors– Embedded system design process –Design example: Model train controller- Design methodologies- Design flows - Requirement Analysis – Specifications-System analysis and architecture design – Quality Assurance techniques - Designing with computing platforms – consumer electronics architecture – platform-level performance analysis.

UNIT-II ***(10 hours)***

Embedded Programming: Components for embedded programs- Models of programs- Assembly, linking and loading – compilation techniques- Program level performance analysis – Software performance optimization – Program level energy and power analysis and optimization – Analysis and optimization of program size- Program validation and testing.

UNIT-III ***(08 hours)***

Real Time Systems: Structure of a Real Time System — Estimating program run times – Task Assignment and Scheduling – Fault Tolerance Techniques – Reliability, Evaluation – Clock Synchronization.

UNIT-IV ***(15 hours)***

Processes and Operating Systems: Introduction – Multiple tasks and multiple processes – Multirate systems-Preemptive real time operating systems-Priority based scheduling- Interprocess communication mechanisms – Evaluating operating system performance- power optimization strategies for processes – Example Real time operating systems-POSIX-Windows CE. - Distributed embedded systems – MPSoCs and shared memory multiprocessors. – Design Example - Audio player, Engine control unit – Video accelerator.

Essential Readings:

1. Marilyn Wolf, “Computers as Components - Principles of Embedded Computing System Design”, Third Edition, Morgan Kaufmann Publisher (An imprint from Elsevier), 2012.

2. Jane W. S. Liu, “Real Time Systems”, Pearson Education, Third Indian Reprint, 2003.
3. Lyla B. Das, “Embedded Systems: An Integrated Approach”, Pearson Education, 2013.

Suggested Readings:

1. Jonathan W. Valvano, “Embedded Microcomputer Systems Real Time Interfacing”, Third Edition Cengage Learning, 2012.
2. David. E. Simon, “An Embedded Software Primer”, 1st Edition, Fifth Impression, Addison-Wesley Professional, 2007

Practical Component: ***(30 hours)***

List of Experiments:

Conduct the following experiments on an ARM CORTEX M3 evaluation board to learn Assembly Language Program and using evaluation version of Embedded 'C' & Keil uVision-4 tool/compiler

1. Write a program to find the factorial of a number.
2. Write a program to add an array of 16 bit numbers and store the 32 bit result in internal RAM.
3. Write a program to find the square of a number (1 to 10) using a look-up table.
4. Write a program to find the largest or smallest number in an array of 32 numbers.
5. Write a program to arrange a series of 32 bit numbers in ascending/descending order.
6. Write a program to count the number of ones and zeros in two consecutive memory locations.
7. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
8. Interface a DAC and generate Triangular and Square waveforms.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Process Control (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Process Control	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To understand the fundamentals of non-linear systems.
2. To understand the modeling concepts of non-linear systems.
3. To understand the stability of non-linear systems.
4. To acquire the knowledge of various non-linear control techniques.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand the basic design techniques in process control.
2. Understand different types of controllers used in process control.
3. Understand the various processes and their control in industries.
4. Develop the knowledge of advanced control strategies with their applications.

UNIT-I ***(15 hours)***

Sensors and Transducers: Basic concepts and working principles of sensors and transducers for measuring process variables like pressure, temperature, level and flow; electromechanical, capacitive, inductive, resistive and photoelectric type proximity sensors.

Concepts of System and Process: Response of first order systems including transfer function and transient response to different forcing functions, introduction to process control loop and salient components, process control terminology, modeling basics, degree of freedom, mass balance, energy Balance equations, linearization of nonlinear systems, modeling of level tank system, continuous Stirred Tank Heater, continuous stirred Tank Reactor, transfer function.

UNIT-II ***(12 hours)***

Controller Principles: Process characteristics, Process Characteristics, Constraints in implementation of process control, terms used in process control, process measurement, control system parameters, discontinuous controller modes, continuous controller modes, composite control modes.

Analog Controllers: General features, electronic controllers, pneumatic controllers, design considerations.

Digital Controllers: Digital simulation of control systems, microprocessor-based controller.

UNIT-III ***(10 hours)***

Control Loop Characteristics: Control system configuration, multivariable control system, control system quality and stability, process loop tuning.

Control Equipment and Final Control Elements: Details of controllers including measurement unit, comparator, actuator and final control elements; pneumatic, hydraulic and electronic actuators; control valve characteristic, pneumatic to electric and electric to pneumatic converters, hydraulic and pneumatic power supply system.

UNIT-IV ***(08 hours)***

Advanced control strategies with case studies: Use of DDC and PLC, introduction to supervisory control, conversion of existing control schemes in operating plants, data loggers.

Essential Readings:

1. Process Control by Harriott Peter, Tata McGraw-Hill Publishing Company Limited, 2008.
2. Process Controls: Principles and Applications by S. Bhanot, Oxford Higher Educations, 2008.
3. Process control: a practical approach. John Wiley & Sons, Myke King, 2016.

Suggested Readings:

1. Process Systems Analysis and Control by Coughanowr and LeBlanc, Third Edition, McGrawHill , 2009.
2. Process Control Instrumentation Technology by C.D. Johnson, 8thEd.,Prentice Hall of India Private Limited, 2008.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Design and Simulation of Sampled Data Control System with Deadbeat Controller.
2. Design and Simulation of Sampled Data Control System with Dahlin's Controller.
3. Implementation of Cascade Control System.
4. Dead Time Compensation Using Smith Predictor Algorithm Simulation Using Simulink.
5. Design and Testing Of Inverse Response Compensator Using Simulink.
6. Air Temperature Control System.
7. Study of Labview Software.
8. Simulation of a temperature process Using Labview.
9. Study of Programmable Logic Controller.
10. Direction control of a DC Motor Using Programmable Logic Controller.
11. Process Identification using Least Square Estimation.
12. Understanding FOPTD and SOPTD modeling of systems with MATLAB.
13. PC based control of simulated process.
14. Study of Fuzzy Tool Box and design of Fuzzy Logic Controller for a given process.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Smart Energy Storage System (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Smart Energy Storage System	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To understand the need, characteristics, and techno-economic role of energy storage in modern power systems.
2. To study advanced energy storage technologies and their performance metrics.
3. To analyze smart integration of storage with grids, microgrids, and renewable energy systems.
4. To introduce control, communication, and management architectures for intelligent energy storage operation.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the role of smart energy storage in renewable-rich power systems.
2. Classify and compare different energy storage technologies based on technical and operational criteria.
3. Analyze storage deployment for grid, microgrid, and electric mobility applications.
4. Understand control, monitoring, and aggregation of distributed storage systems.

UNIT-I ***(15 hours)***

Fundamentals of Smart Energy Storage Systems: Characteristics of electrical energy and challenges in power system operation, Need for energy storage in conventional and renewable-based systems, Peak demand, grid congestion, intermittency, and reliability issues, Evolution from conventional storage to smart energy storage systems, Role of storage in smart grids, microgrids, and prosumer-based energy systems, Key performance parameters: power rating, energy capacity, efficiency, response time, lifecycle, and cost.

UNIT-II ***(12 hours)***

Energy Storage Technologies and Classification: Overview and classification of energy storage systems, Mechanical Storage: Pumped Hydro Storage (PHS), Compressed Air Energy Storage (CAES), Flywheel Energy Storage (FES), Electrochemical Storage: Lead-acid, Lithium-ion, Sodium-sulfur, Flow batteries, Electrical Storage: Supercapacitors (DLC), Superconducting Magnetic Energy Storage (SMES), Chemical and Thermal Storage: Hydrogen energy storage, Synthetic Natural Gas (SNG), thermal energy storage, Comparative assessment of technologies for smart applications.

UNIT-III ***(10 hours)***

Smart Integration, Control, and Standards: Integration of storage with renewable energy sources (solar, wind), Storage roles from utility, consumer, and renewable generator perspectives, Power electronics interfaces for energy storage systems, Battery Management Systems (BMS) and Energy Management Systems (EMS), Communication, monitoring, and control architectures, Standards, safety, and grid codes for smart energy storage deployment.

UNIT-IV ***(8 hours)***

Applications and Advanced Concepts in Smart Energy Storage: Utility-scale applications: frequency regulation, peak shaving, load leveling, Storage in smart grids and smart microgrids, Residential and commercial storage: smart homes and buildings, Energy storage for electric vehicles and charging infrastructure, Emerging trends and future research directions in smart energy storage.

Essential Readings:

1. D. Linden and T. B. Reddy, *Handbook of Batteries*, 4th Edition, McGraw-Hill, 2011.
2. Ibrahim Dincer and Marc A. Rosen, *Thermal Energy Storage: Systems and Applications*, 2nd Edition, Wiley, 2010.
3. Sean Daly, Drew Lebowitz, and Swetha Sundaram, *The BESS Book: A Cell to Grid Guide to Utility-Scale Battery Energy Storage Systems* (Bess Book, 2024).

Suggested Readings:

1. Xisheng Tang, Zhiping Qi , Li Kong, *Electrical Energy Storage Technologies and Applications*, Springer, 2025.
2. A. A. Akhil, G. Huff, A. B. Currier, et al., *Energy Storage Systems: Fundamentals, Classification and a Technical Comparative*. Albuquerque, NM, USA: Sandia National Laboratories, 2013.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To study charge–discharge characteristics of a battery energy storage system and evaluate its efficiency and state of charge.
2. To model and analyze lithium-ion battery behavior using equivalent circuit models under dynamic operating conditions
3. To study battery management system functions, including state of charge estimation, state of health assessment, and cell balancing.
4. To analyze the integration of energy storage systems with a solar PV system for power smoothing and peak shaving.
5. To model and analyze high-power energy storage systems such as flywheels and supercapacitors for short-duration applications.
6. To study the operation of energy storage systems in grid-connected and islanded microgrid modes.
7. To analyze thermal and safety aspects of battery energy storage systems under different operating conditions.
8. To analyze the techno-economic feasibility of energy storage systems for renewable energy integration.

(Note: The Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Wind and Small Hydro Power Generation (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wind and Small Hydro Power Generation	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. Understand wind and small hydro fundamentals, resource assessment, and site selection.
2. Analyze wind turbine aerodynamics, generators, and farm design.
3. Learn small hydro classification, hydrology, turbines, and components.
4. Evaluate hybrid integration, grid challenges, environmental, and economic aspects.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. To appraise basic principles of wind energy conversion, various wind parameter measuring instruments and to estimate potential of wind energy resource
2. To appraise concept of aerodynamics geometry, design and performance of HWAT and concepts of VAWT
3. To make calculations of the economics of the wind energy system.
4. To appraise small hydro system components and design, hybrid systems and controls.

UNIT-I ***(12 hours)***

Introduction to Renewable Energy and Wind Power Basics

Overview of renewable energy sources: Global and Indian energy scenario, potential, advantages, and limitations of wind and hydro power. Wind energy fundamentals: Nature of wind, wind data analysis (Weibull distribution), power in the wind, Betz limit, wind resource assessment, site selection criteria. Wind energy program in India: Status, potential sites, and policies. Types of wind turbines: Horizontal axis (HAWT) and vertical axis (VAWT), components (blades, rotor, gearbox, generator, tower).

UNIT-II ***(15 hours)***

Wind Energy Conversion Systems

Aerodynamics of wind turbines: Blade design, number of blades, pitch control, tip speed ratio (λ), power coefficient (Cp-λ curves). Wind turbine generators: Fixed-speed vs. variable-speed systems, induction generators, synchronous generators, doubly-fed induction generators (DFIG). Performance characteristics: Power curve, capacity factor, cut-in/cut-out speeds, efficiency. Wind farm design: Layout, wake effects, offshore wind energy, environmental impacts, and economic aspects.

UNIT-III ***(10 hours)***

Small Hydro Power Fundamentals

Introduction to hydropower: Classification (large, small, micro, mini, pico hydro), potential in India, advantages, and limitations. Hydrology basics: Resource assessment, flow duration curves, head and flow rate measurement, site selection. Components of small hydro systems: Intake, penstock, turbines, generators, tailrace, and control systems. Types of turbines for small hydro: Pelton (impulse), Francis (reaction), Kaplan, propeller, cross-flow; selection criteria based on head and flow.

UNIT-IV

Small Hydro Power Systems and Integration ***(8 hours)***

Performance and efficiency: Effect of head, flow rate, and load on power output; specific speed, voltage regulation. Small hydro plant design: Civil works, speed and voltage control, standalone vs. grid-connected systems. Environmental and economic aspects: Impacts on ecology, siltation issues, cost-benefit analysis, tariffs, and subsidies in India. Hybrid systems: Integration of wind and small hydro with other renewables (e.g., solar), energy storage, and grid interconnection challenges.

Essential Readings:

1. J. F. Manwell, J. G. McGowan, and A. L. Rogers, Wind Energy Explained: Theory, Design and Application, John Wiley & Sons Ltd., 2010.
2. L. L. Freris, Wind Energy Conversion Systems, Prentice Hall, 1990.
3. Martin O. L. Hansen, Aerodynamics of Wind Turbines, Earthscan, 2008.
4. Adam Harvey, Micro-Hydro Design Manual: A Guide to Small-Scale Water Power Schemes, Intermediate Technology Publications, 1993.

Suggested Readings:

1. J. F. Walker, *Wind Energy Technology*, John Wiley, 1997.
2. G. L. Johnson, *Wind Energy Systems*, Prentice Hall, 1985.
3. Roman Ritter, *Good and Bad of Mini Hydro Power*, 2009.

Practical Components: ***(30 hours)***

List of Experiments:

1. Wind measurement for a month and Statistical Analysis of Wind data.
2. To study the construction and working of various types of HAWT.
3. To study the construction and working of various types of VAWT.
4. To study the wind farm planning and development.
5. Case study on wind energy economics.
6. Study of the generator used in wind power plants (PMDC / Induction generator.
7. To draw the performance curves of a horizontal-axis wind turbine using lamps as an electrical load.
8. To draw the performance curves of a horizontal-axis wind turbine using a battery (Accumulator) as an electrical load.
9. To study and design of small hydro turbine.
10. To study and design mini and micro hydro turbines.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Embedded Based Smart System Design (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Embedded Based Smart System Design	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To introduce the fundamental concepts and architecture of embedded systems used in automotive engineering, with emphasis on electric and hybrid vehicles.
2. To enable students to design and develop embedded solutions for vehicle control, communication, diagnostics, and safety systems.
3. To provide hands-on experience in programming, testing, and integrating embedded hardware and software for automotive applications.
4. To develop the ability to analyze performance, reliability, and safety requirements of embedded systems deployed in automotive environments.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain embedded system fundamentals and their application in automotive and electric vehicle subsystems.
2. Design and implement embedded programs for vehicle control and monitoring applications.
3. Interface sensors, actuators, and communication modules used in automotive embedded systems.
4. Analyze and evaluate the performance, reliability, and safety aspects of embedded systems in automotive environments.
5. Integrate multidisciplinary knowledge to develop embedded solutions for real-world challenges in vehicle automation and smart mobility.

UNIT-I ***(10 hours)***

Introduction to Embedded Systems in EHV: Overview of embedded systems and their role in automotive engineering, electric and hybrid vehicle system architecture, role of embedded systems in EV subsystems: battery management system (BMS), vehicle control unit (VCU), motor control unit and charger controller, fundamentals of microcontrollers and microprocessors used in automotive applications, automotive-grade embedded hardware requirements, introduction to real-time operating systems (RTOS), importance of RTOS in safety-critical automotive and EV applications.

UNIT-II ***(12 hours)***

Automotive Communication Protocols: Need for in-vehicle communication networks, overview of automotive communication protocols: CAN bus, LIN bus, and FlexRay, CAN frame structure and arbitration mechanism, applications of CAN communication in EHV, embedded communication between BMS, VCU, and motor controller.

UNIT-III ***(12 hours)***

Embedded Systems for Vehicle Diagnostics and Maintenance: Introduction to onboard diagnostics (OBD) systems, OBD-II architecture and standards, overview of ISO 15765 CAN-based diagnostics, embedded monitoring of vehicle performance, battery health, and fault conditions, diagnostic trouble code (DTC) generation, retrieval, and interpretation, introduction to remote diagnostics and over-the-air (OTA) software updates in modern electric vehicles.

UNIT-IV ***(11 hours)***

Automotive Infotainment, Human–Machine Interface, and Vehicle Safety: Introduction to automotive infotainment systems and human–machine interface (HMI) design principles, embedded displays, warning indicators, and driver information systems, overview of advanced driver assistance systems (ADAS), embedded systems for collision warning, lane departure warning, and adaptive cruise control, case studies of embedded system applications in vehicle safety and driver assistance systems.

Essential Readings:

1. R. Zurawski, Embedded Systems Handbook: Embedded systems design and verification, CRC press, 2018.
2. J. Wang, Real-time embedded systems, John Wiley & Sons, 2017.
3. R. Zurawski, Embedded Systems Handbook 2-Volume Set, CRC press, 2018.

Suggested Readings:

1. M. Kathiresh, R. Neelaveni, Automotive Embedded Systems, Springer International Publishing, 2021.
2. N. Subhashini, K. Mohanaprasad, M. Murugan, Intelligent embedded systems, D. Thalmann (Ed.). Springer, 2018.
3. G. Nicolescu, P. J. Mosterman, Model-based design for embedded systems, CRC Press, 2018.

Practical Components: ***(30 hours)***

List of Experiments:

1. Study of Automotive Embedded System Architecture.
2. Microcontroller Programming for Automotive Applications.
3. Battery Voltage and Current Measurement Using ADC.
4. Battery Temperature Monitoring and Protection System.
5. PWM-Based DC/BLDC Motor Speed Control.
6. Relay and Contactor Control for EV Power Management.
7. CAN Bus Communication Between Automotive ECUs.
8. On-Board Diagnostics (OBD) and Fault Code Generation.
9. State of Charge (SOC) Estimation Using Embedded System.
10. Embedded HMI for Electric Vehicle Monitoring.
11. Mini Embedded Project for Electric/Hybrid Vehicle Application.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Computer Aided Design of Electric Machines (DSE-09/ GE-09)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Computer Aided Design of Electric Machines	4	3	0	1	Electrical Machines I, Electrical Machines II

Course Objectives:

1. To develop a strong analytical understanding of modern electric motor topologies used in electric vehicle traction.
2. To enable students to mathematically model torque production, magnetic circuits, and electromechanical energy conversion in advanced electric motors.
3. To train students in step-by-step electromagnetic design methodology of EV traction motors considering performance, thermal, and mechanical constraints.
4. To expose students to computer-aided design, FEM-based analysis, and optimization techniques for modern electric motors through case studies.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Analyze and compare advanced electric motors for electric vehicle applications based on torque density, efficiency, and control requirements.
2. Develop mathematical models and equivalent circuits of BLDC, PMSM, SynRM, and SRM using dq-axis theory, co-energy method, and nonlinear magnetic modeling.
3. Design EV traction motors from given specifications by calculating magnetic, electrical, thermal, and mechanical parameters using analytical design procedures.
4. Apply FEM tools to evaluate magnetic field distribution, torque, losses, and saturation effects in electric machines.
5. Optimize machine performance using parametric studies and basic optimization techniques while interpreting practical design trade-offs in EV applications.

UNIT-I ***(12 hours)***

Design of Synchronous Reluctance: Introduction and role of reluctance machines in EV traction, comparison of EV motors, torque production mechanisms, dq-axis modeling, saliency ratio and reluctance torque, equivalent circuit representation, power drive motor design requirements, step-by-step design methodology, EV traction case study.

Design of PM-Assisted Motors: Motivation for PM-assisted SynRM, magnetic saturation effects, permanent magnet placement strategies, linear start motor design, parameterization.

UNIT-II ***(12 hours)***

Design of Brushless Permanent Magnet Motors: Historical evolution, applications in electric vehicles, aerospace and industrial drives, permanent magnet materials and characteristics, magnetic circuit modeling, demagnetization and thermal considerations.

Design of Permanent Magnet Synchronous Motors: Configurations and mathematical modeling, fundamental design relationships, winding design and EMF equations, design scope and constraints, optimal design with case studies.

Design of Brushless DC Motors: Mathematical modeling, surface-mounted motor construction and design, design guidelines, numerical problem on motor design.

UNIT-III *(11 hours)*

Design of Switched Reluctance Motors: Introduction, applications in EVs and harsh environments, construction and operating principle, phase voltage equations, electromagnetic torque equations, nonlinear inductance modeling, co-energy method for torque computation, analytical flux linkage computation, air-gap permeance modeling, aligned and unaligned inductance calculation, design specifications, stator design parameters, rotor pole design, winding design and current profiling, performance calculation, mechanical stress considerations, thermal design considerations, design workflow from specifications to performance evaluation.

UNIT-IV *(10 hours)*

Advanced Motors: Axial flux motors: Introduction, torque and force equations, performance characteristics, design scope; axial flux switched reluctance motors (AFSRM): introduction, applications and classification, operating principle, analytical design equations.

computer aided designing and analysis method (Case-Study): FEM-based design methodology, FEM problem formulation for EV motors, critical simulation settings for FEA, meta model-based simulation optimization, overview of optimization algorithms: genetic algorithm and particle swarm optimization.

Essential Readings:

1. A. K. Sawhney, A Course in Electrical Machine Design, Dhanpat Rai and Co., 2016.
2. J. R. Hendershot and T. J. E. Miller, Design of Brushless Permanent – Magnet Motors, Motor Design Books LLC, 2nd edition, 2010.
3. R. Krishnan, Switched Reluctance Motor Drives, CRC Press LLC, USA, 2001.

Suggested Readings:

1. T. A. Lipo, Introduction to AC Machine Design, IEEE Press – Wiley Publications, 2017.
2. J. Pyrhonen, T. Jokinen, and V. Hrabovcova, Design of Rotating Electrical Machines, John Wiley and Sons Inc., 2nd edition, 2013.

Practical Components: ***(30 hours)***

List of Experiments:

1. Performance comparison of PMSM, SynRM, and SRM based on torque–speed and efficiency characteristics.
2. Computation of reluctance torque using saliency ratio and current angle variation in a synchronous reluctance motor.
3. Analytical design of a synchronous reluctance motor for EV traction based on given power and speed specifications.
4. Analysis of torque enhancement and saturation using different PM placement strategies in a synchronous reluctance motor.
5. Calculation of air-gap flux density and demagnetization limits of permanent magnet motors.
6. Determination of PMSM winding turns, back-EMF constant, and torque constant.
7. Simulation of BLDC motor phase currents, back-EMF, and electromagnetic torque.
8. Analytical calculation of surface-mounted BLDC motor dimensions, torque, and losses.
9. Flux linkage and torque computation of SRM using the co-energy method.
10. Determination of switched reluctance motor stator and rotor pole arcs, inductances, and torque profile.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Clean Energy Technologies (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Clean Energy Technologies	4	3	1	0	Introduction to Electrical and Electronics Engineering, Physics

Course Objectives:

1. To introduce clean energy concepts and the need for sustainable technologies.
2. To explain clean production from fossil fuels and bioenergy sources.
3. To describe solar and wind energy systems and their applications.
4. To cover other renewable technologies and economic/environmental aspects.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understand the principles of clean energy and its importance in sustainability.
2. Analyze clean coal technologies and biofuel production processes.
3. Explain solar photovoltaic and thermal systems, including performance calculations.
4. Describe wind, hydro, and geothermal energy conversion and evaluate economic viability.

UNIT-I ***(11 hours)***

Introduction to Clean Energy: Global energy scenario, environmental impacts of fossil fuels, need for clean technologies; Clean coal technologies: gasification, fluidized bed combustion, carbon capture and storage (CCS); Supercritical and ultra-supercritical power plants.

UNIT-II ***(11 hours)***

Bioenergy Technologies: Biomass resources, thermochemical conversion (pyrolysis, gasification); Biochemical conversion (anaerobic digestion, fermentation); Biofuels: biodiesel production (transesterification), bioethanol; Biogas plants and applications.

UNIT-III ***(10 hours)***

Solar Energy Technologies: Solar radiation, measurement, and models; Solar thermal systems - collectors, concentrators, solar water heating, power cycles; Solar photovoltaic: cell types, efficiency, PV modules, inverters, grid integration.

UNIT-IV ***(13 hours)***

Wind and Other Renewables: Wind energy: turbine types, power curves, Betz limit, site assessment; Hydroelectric: small hydro, tidal, and ocean energy basics; Geothermal energy: resources, binary cycles; Economic analysis: levelized cost of energy, payback period; Policy and sustainability issues.

Essential Readings:

1. Godfrey Boyle, Renewable Energy: Power for a Sustainable Future, Oxford University Press, 3rd Edition, 2012.
2. Bent Sorensen, Renewable Energy: Physics, Engineering, Environmental Impacts, Economics and Planning, Academic Press, 5th Edition, 2017.
3. Chetan Singh Solanki, Renewable Energy Technologies: A Practical Guide for Beginners, PHI Learning, 2019.

Suggested Readings:

1. G.D. Rai, Non-Conventional Energy Sources, Khanna Publishers, 5th Edition, 2010.
2. John Twidell and Tony Weir, Renewable Energy Resources, Routledge, 3rd Edition, 2015.
3. Sukhatme and Nayak, Solar Energy: Principles of Thermal Collection and Storage, McGraw Hill Education, 4th Edition, 2018.

Tutorial Component: ***(15 Hours)***

List of Tutorial Tasks:

1. Comparison of environmental impacts and efficiency of conventional vs. clean coal technologies (gasification, FBC, CCS).
2. Calculation of efficiency improvement in supercritical/ultra-supercritical plants using Rankine cycle modifications.
3. Selection criteria for site and technology in clean coal power plants (case-study discussion).
4. Classification and comparison of thermochemical vs. biochemical biomass conversion routes with yield calculations.
5. Design considerations and performance evaluation of biogas plants (digester sizing and methane yield).
6. Biodiesel/bioethanol production process flow and transesterification efficiency calculations.
7. Solar radiation measurement, tilt-angle optimization, and insolation calculations for a given location.
8. Performance analysis and efficiency comparison of flat-plate vs. concentrating solar thermal collectors.
9. PV module sizing, I-V curve interpretation, and grid integration challenges (sketching and discussion).
10. Wind turbine power curve analysis, Betz limit application, and site assessment using wind data.
11. Classification and basic power output calculations for small hydro, tidal, and geothermal systems.
12. Economic evaluation (LCOE, payback period) and policy comparison of different clean energy technologies (group discussion/case study).

(Note: The course instructor may add/delete/update new tutorial tasks in addition to the above suggested tutorial tasks.)

Introduction to Smart Grid (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Smart Grid	4	3	1	0	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To introduce the concept of smart grid and its evolution from traditional power systems.
2. To explain the architecture, components, and enabling technologies of smart grids.
3. To describe communication infrastructure and standards in smart grids.
4. To familiarize students with applications, renewable integration, microgrids, and challenges in smart grid implementation.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Differentiate between conventional grids and smart grids, and explain the need for smart grids.
2. Describe the functional components and architecture of smart grids.
3. Understand communication technologies and standards used in smart grids.
4. Analyze applications of smart grids in renewable integration, microgrids, and demand-side management.

UNIT-I ***(10 hours)***

Introduction to Smart Grids: Evolution of electric power systems; Limitations of traditional grids; Concept and definition of smart grid; Drivers and need for smart grids; Smart grid functions and benefits; Opportunities and challenges in smart grid implementation.

UNIT-II ***(14 hours)***

Smart Grid Architecture and Components: Conceptual model and domains (generation, transmission, distribution, customer); Key components: advanced metering infrastructure (AMI), phasor measurement units (PMU), wide area monitoring systems (WAMS); Distributed energy resources (DER) integration; Energy storage systems.

UNIT-III ***(10 hours)***

Communication and Standards: Communication technologies in smart grids: wired (PLC, optical fiber) and wireless (ZigBee, Wi-Fi, cellular); IEC 61850 standard for substation automation; Cyber security basics in smart grids; Interoperability and standards.

UNIT-IV ***(11 hours)***

Applications and Advanced Concepts: Renewable energy integration in smart grids; Microgrids: AC/DC types, operation modes (grid-connected, islanded); Demand-side management and demand response; Electric vehicles and vehicle-to-grid (V2G); Smart grid initiatives in India.

Essential Readings:

1. James Momoh, Smart Grid: Fundamentals of Design and Analysis, Wiley-IEEE Press, 2012.
2. Janaka Ekanayake, Nick Jenkins, Kithsiri Liyanage, Jianzhong Wu, Akihiko Yokoyama, Smart Grid: Technology and Applications, Wiley, 2012.

3. Clark W. Gellings, The Smart Grid: Enabling Energy Efficiency and Demand Response, Fairmont Press, 2009.

Suggested Readings:

1. Stuart Borlase, Smart Grids: Infrastructure, Technology, and Solutions, CRC Press, 2nd Edition, 2017.
2. Tony Flick and Justin Morehouse, Securing the Smart Grid: Next Generation Power Grid Security, Syngress, 2010.
3. Vehbi C. Güngör and Dilan Sahin, Smart Grid Communications and Networking, Cambridge University Press, 2012.

Tutorial Component: ***(15 Hours)***

List of Tutorial Tasks:

1. Comparison of conventional grids vs. smart grids (features, limitations, and benefits).
2. Identification of drivers, opportunities, and challenges in smart grid adoption (case-study discussion).
3. Listing smart grid functions and benefits with real-world examples.
4. Conceptual modelling of smart grid domains and key component identification.
5. Sizing and application of AMI, PMU, and WAMS in sample power systems.
6. Integration of DER and energy storage systems in smart grid architecture (sketching and discussion).
7. Selection of communication technologies (wired vs. wireless) for different smart grid layers.
8. Interpretation of IEC 61850 standards in substation automation scenarios.
9. Basic cyber-security risk assessment and interoperability issues in smart grid communication.
10. Renewable energy integration and microgrid operation modes (grid-connected vs. islanded).
11. Demand response strategies, V2G calculations, and electric vehicle integration.
12. Economic and policy analysis of smart grid initiatives (group discussion/case study).

(Note: The course instructor may add/delete/update new tutorial tasks in addition to the above suggested tutorial tasks.)

Optimal and Robust Control (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Optimal and Robust Control	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I.

Course Objectives:

1. To acquire the fundamental knowledge of optimal control framework.
2. To understand the designs of optimal control techniques.
3. To acquire the fundamental knowledge of robust control theory.
4. To understand the robust control techniques.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Design and analyze the closed loop system with linear state feedback control laws.
2. Design the observer based controllers.
3. Analyze robustness of the control systems.
4. Design robust control algorithms on the engineering systems.

UNIT-I ***(08 hours)***

Introduction to Optimal Control: Introduction to optimal control issues and framework, state feedback for regulation and tracking, observer based compensators and separation property, steady state tracking with observer based compensators, related examples.

UNIT-II ***(12 hours)***

Optimal control general mathematical procedures: Formulation of optimal control problem, performance indices, calculus of variation, minimum principle, dynamic programming, related examples, optimal feedback control linear state regulator, continuous time linear state regulator, use of linear regulator to solve other linear optimal control problems, minimum time control of LTI systems, stability analysis, related examples.

UNIT-III ***(15 hours)***

Introduction to robust control: Vector spaces, linear subspaces, invariant subspaces, vector norms and matrix norms, singular value decomposition, semi-definite matrices, description of linear systems, operations on systems, state space realizations for transfer matrices.

Feedback interconnection & stability: Well-posedness, internal stability, co-prime factorization, stabilizing controllers, Linear Quadratic Regulators: Return ratio and difference, design of LQR and robustness analysis.

UNIT-IV ***(10 hours)***

Other Robust Control Techniques: LQG/LTR Design and its robustness analysis, related examples, H_2 and $H\infty$ control techniques and their design formulations.

Essential Readings:

1. Modern Control System Theory by M Gopal, New Age International (P) Limited, 2015.

2. Linear State-space Control Systems by Robert L Williams II & Douglas A. Lawrence, John Wiley & Sons, INC, 2007.
3. Essentials of Robust Control by Kemin Zhou and J C Doyle, Prentice Hall, 1998.

Suggested Readings:

1. Multivariable Feedback Design by J M Maciejowski, Addison-Wesley Publishing Company, 1989.
2. Linear Optimal Control by B D O Anderson and J B Moore, Prentice Hall, 1990.

Practical Component: ***(30 Hours)***

List of Experiments:

1. State Space realization of SISO system and implementation of optimal control for the same.
2. State Space realization of MIMO system and implementation of optimal control for the same.
3. To obtain the various norms of matrix, signals, vectors and systems.
4. To implement LQR control for SISO system.
5. To implement LQR control for MIMO system.
6. To implement LQG/LTR control for SISO system.
7. To implement LQG/LTR control for MIMO system.
8. To implement $H\infty$ control technique for SISO system.
9. Applications of robust control techniques in real problems.
10. Case Study1: Implementation of optimal control for any practical application.
11. Case Study2: Implementation of robust control for any practical application.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Robotics in Biomedical Engineering (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Robotics in Biomedical Engineering	4	3	1	0	Introduction to Robotics and Mechatronics

Course Objectives:

1. To explore the medical robots and their uses in rehabilitation purpose.
2. To investigate recent developments and emerging trends in biomedical robots and interpret localization and navigation techniques for mobile robots.
3. To identify the fundamental parameters of medical and mobile robots and examine their typical applications.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Design and implement robotic assistants for Minimally Invasive Surgery and Image-Guided Interventions and analyze key design parameters of medical robots.
2. Evaluate the parameters of rehabilitation systems, exoskeletons, and explore locomotion aspects of mobile robots.
3. Analyze the core applications of specialized surgical robots and assess their advantages and limitations.

UNIT-I ***(12 hours)***

Introduction to Medical Robots: Introduction to Bio-medical robots, Types of Medical Robots, Navigation and paradigms of BioMedical Robots, Forward kinematics, Inverse, Typical applications and benefits of robots in health care sectors. Design of Medical Robots, Characterization of gestures to the design of robot-design methodology- technological choices-security

UNIT-II ***(12 hours)***

Rehabilitation Robotics: Introduction, Exoskeletons-Design concepts, Development and control-Human hand-Biomechanics, Rehabilitation for Limbs-Brain-Machine-Interfaces, Redundancy resolution, Introduction to Rehabilitation Strategies, Robotic prosthetics. Mobile Robot locomotion: Types of locomotion and its salient characteristics of hopping robots, legged robots, wheeled robots, stability, aerial robots, maneuverability, controllability.

UNIT-III ***(11 hours)***

Mobile Robot Localization and Navigation: Introduction, the challenges of localization, localization based navigation versus programmed solutions, map representation, probabilistic map, map based localization, autonomous map building. Planning and navigation: Planning and reaction, D* algorithm obstacle avoidance.

UNIT-IV ***(10 hours)***

Current Topics in Bio-Medical Robotics: Haptic Augmentation in Exoskeletons, Robotic Catheters for percutaneous interventions, Unsupervised learning for mapping in Bio-Robots, Reven–II Robots. Future Trends of Robotics in the Medical Field.

Essential Readings:

1. Paula Gomes, "Medical robotics- Minimally Invasive surgery", Woodhead Publishing, 2012
2. Jocelyne Troccaz, "Medical Robotics", Wiley-ISTE, 2012.
3. Vanja Bonzovic, "Medical Robotics", I-tech Education publishing, Austria, 2008.

Suggested Readings:

1. Daniel Faust, "Medical Robots", Rosen Publishers, 2016.
2. Jacob Rosen, Blake Hannaford & Richard M Satava, "Surgical Robotics: System Applications & Visions", Springer 2011.

Tutorial Component: ***(15 Hours)***

(Note: The course instructor will design tasks to complete the tutorial component of the course.)

Smart Grid Technologies (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Smart Grid Technologies	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To introduce the fundamental concepts, architecture, standards, and enabling technologies of Smart Grid systems.
2. To understand distributed generation resources, energy storage technologies, and battery management systems for Smart Grid applications.
3. To study wide-area monitoring, phasor measurement, protection schemes, and islanding detection techniques in Smart Grids.
4. To analyze the operation, control, and energy management of AC, DC, and hybrid AC–DC microgrids with practical case studies.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain Smart Grid concepts, architectures, standards, cybersecurity issues, and demand-side management strategies.
2. Analyze distributed generation and energy storage systems, including modelling, sizing, siting, and BMS operation.
3. Apply wide-area monitoring systems, PMU-based phasor estimation, digital protection, and islanding detection techniques in Smart Grids.
4. Analyze the operation, control, and protection of AC, DC, and hybrid microgrids under grid-connected and islanded modes.

UNIT-I ***(15 hours)***

Smart Grid Fundamentals, Architecture, and Enabling Technologies: Introduction to Smart Grid, Architecture of Smart Grid System, Standards of Smart Grid System, Elements and Technologies of Smart Grid System, Cybersecurity and resiliency of Smart Grid, Demand Side Management (DSM) of Smart Grid, Demand Response Analysis of Smart Grid, Virtual inertia and ancillary support in Smart Grid.

UNIT-II ***(10 hours)***

Distributed Generation and Energy Storage Systems: Distributed Generation Resources, Introduction to energy storage devices, Types of energy storage technologies, Analytical modelling of energy storage devices, Modelling of storage devices, Optimal sizing and siting of energy storage systems, Battery Management System (BMS).

UNIT-III ***(10 hours)***

Monitoring, Protection, and Islanding in Smart Grids: Wide Area Monitoring Systems (WAMS), Phasor Estimation, Phasor Measurement Unit (PMU) placement, Digital relays for Smart Grid protection, Islanding Detection Techniques, Smart Grid Protection, Harmonic effects and mitigation techniques.

UNIT-IV ***(10 hours)***

Microgrids, Energy Management, and Advanced Applications: Modelling of DC Smart Grid components, Operation and control of: AC Microgrid, DC Microgrid, AC–DC Hybrid Microgrid, Hierarchical control techniques in hybrid AC–DC microgrids, Energy management design of Smart Grid, System analysis of AC/DC Smart Grid.

Essential Readings:

1. Janaka Ekanayake, Nick Jenkins, Kithsiri Liyanage, and Jianzhong Wu, *Smart Grid: Technology and Applications*, John Wiley & Sons, 1st Edition, 2012.
2. Stuart Borlase (Ed.), *Smart Grid: Infrastructure, Technology, and Solutions*, CRC Press, 1st Edition, 2013.
3. James Momoh, *Smart Grid: Fundamentals of Design and Analysis*, John Wiley & Sons, 1st Edition, 2012.

Suggested Readings:

1. Hatim G. Shaker, Ahmed F. Zobaa, *Wide Area Monitoring, Protection and Control Systems: The Enablers for Smarter Grids*, IET, 1st Edition, 2015.
2. Nikos Hatziargyriou (Ed.), *Microgrids: Architectures and Control*, Wiley–IEEE Press, 1st Edition, 2014.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To model and simulate distributed generation resources in a Smart Grid using software tools.
2. To perform analytical modelling, optimal sizing, and siting of energy storage systems.
3. To simulate phasor estimation techniques and PMU placement in a Smart Grid network.
4. To simulate the operation and control of an AC microgrid under grid-connected and islanded modes.
5. To simulate the operation and control of a DC microgrid and a grid-connected DC microgrid.
6. To analyze harmonic effects in Smart Grids and implement mitigation techniques.
7. To perform a simulation and case study of AC, DC, and AC–DC hybrid microgrids.
8. To demonstrate energy management in a microgrid using real-time or simulation-based platforms.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Energy Management Systems and SCADA (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Energy Management Systems and SCADA	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Objectives:

1. To understand the functions, architectures, and recent developments of Energy Management Centers and their role in power system operation.
2. To analyze power generation characteristics, economic dispatch, unit commitment, generation scheduling, and interchange costing in interconnected power systems.
3. To study the principles, architectures, components, and communication protocols of SCADA systems for power system monitoring and control.
4. To understand the architecture, control functions, and applications of Energy Management Systems (EMS), including AGC, LFC, VQC, security assessment, and dispatch.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Acquire and demonstrate knowledge of energy management centers, power system operation, and generation characteristics.
2. Apply principles of economic dispatch, unit commitment, and generation scheduling, considering operational constraints and regional interchanges.
3. Analyze Energy Management System (EMS) architectures, online and offline functions, and real-time control applications in power systems.
4. Understand and analyze SCADA system components, configurations, communication requirements, and protocols used for power system monitoring and automation.
5. Analyze advanced EMS and SCADA applications such as AGC, LFC, VQC, security assessment, contingency analysis, power flow, and state estimation in modern power systems.

UNIT-I ***(13 hours)***

Energy Management and Generation Scheduling: Energy Management Centers: functions, architectures, and recent developments, Characteristics of power-generating units, Economic dispatch of generating units, Unit commitment: spinning reserve, thermal, hydro, and fuel constraints, Solution techniques for unit commitment, Generation scheduling with limited energy, Energy production cost: cost models, budgeting, planning, and practical considerations, Interchange evaluation for regional power system operations, Types of power interchanges and exchange costing techniques.

UNIT-II ***(12 hours)***

Supervisory Control and Data Acquisition (SCADA) Fundamentals: Introduction to SCADA systems, SCADA functional requirements and components, General features, functions, applications, and benefits of SCADA, SCADA configurations and system architectures, Remote Terminal Units (RTUs): hardware and connections, SCADA communication requirements, SCADA communication protocols: past, present, and future, Structure of a SCADA communication protocol.

UNIT-III ***(10 hours)***

Energy Management Systems (EMS) Architecture and Control: Introduction and evolution of EMS: control centers, energy control centers, and EMS, Functions and benefits of EMS, State

Electricity Boards (SEBs) monitoring and control, EMS architectures and applications, On-line and offline functions of EMS, Real-time modeling and applications of EMS, EMS control functions: Automatic Generation Control (AGC), Load Frequency Control (LFC), Voltage and Reactive Power Control (VQC).

UNIT-IV ***(10 hours)***

SCADA, EMS Applications, and Power System Automation: EMS case studies: Security assessment Economic dispatch, Contingency analysis, Study-mode applications: Forecasting, Power flow and optimal power flow, State estimation, Security assessment, SCADA evolution, architectures, modules, and components, SCADA hardware: RTUs, IEDs, and Substation Automation Systems (SAS), SCADA software and standards: IEC 61850, GOOSE protocol, Applications of SCADA: Power systems, Railways, Renewable energy systems, Smart grid, Power SCADA applications: automation, protection, and relay interoperability

Essential Readings:

1. Handschin, E. "Energy Management Systems", Springer Verlag, 1990.
2. Handschin, E. "Real Time Control of Electric Power Systems", Elsevier, 1972.
3. John D McDonald, "Electric Power Substation Engineering", CRC press, 2001.

Suggested Readings:

1. Wood, A. J, and Wollenberg, B. F, "Power Generation Operation and Control", 2nd Edition, John Wiley and Sons, 2003.
2. Green, J. N, Wilson, R, "Control and Automation of Electric Power Distribution Systems", Taylor and Francis, 2007.
3. Turner, W. C, "Energy Management Handbook", 5th Edition, 2004.

Practical Component: ***(30 Hours)***

List of Experiments:

1. To study the Energy Management Center (EMC) architecture and functions using simulation tools.
2. Simulation of the economic dispatch problem for thermal generating units without transmission losses.
3. Simulation of economic dispatch with transmission losses using the penalty factor method.
4. Unit commitment analysis considering spinning reserve and generator constraints.
5. Evaluation of energy production cost using different generator cost models.
6. Simulation of Automatic Generation Control (AGC) for single-area and multi-area power systems.
7. Simulation and analysis of Load Frequency Control (LFC) and Voltage Reactive Power Control (VQC).

(Note: The Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

EV Charging Infrastructure (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
EV Charging Infrastructure	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. Understand the architecture of EV charging infrastructure and applicable Indian and international charging standards.
2. Analyze and design AC–DC and DC–DC power converter topologies with appropriate control strategies for efficient EV charging applications.
3. Understand and apply communication protocols, testing methodologies, and EMI/EMC considerations essential for reliable and compliant EV charging systems.
4. Develop the ability to apply theoretical concepts to practical EV charger design problems and real-world deployment scenarios, including Indian charging ecosystem case studies.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain EV charging infrastructure, EVSE types, charging modes, and Indian and international charging standards.
2. Analyze and simulate AC–DC PFC converters for single-phase and three-phase EV chargers.
3. Design and evaluate isolated DC–DC converters with soft-switching techniques for EV charging applications.
4. Implement and analyze communication protocols and testing methodologies used in EV chargers and EVSE.
5. Evaluate EMI/EMC issues and apply mitigation techniques in EV charging systems.

UNIT-I ***(10 hours)***

Introduction: introduction to EV systems, EV benefits, battery charging modes, types of EV supply equipment (EVSE), components of EV battery chargers, charging infrastructure challenges.

Charger Classification and standards: classification based on charging levels (region-wise), modes, plug types, architecture and specifications of Bharat AC001 and DC001 chargers, standards related to: connectors, communication, supply equipments, electromagnetic interference (EMI)/ electromagnetic compatibility (EMC).

UNIT-II ***(12 hours)***

AC-DC Converter: types of AC-DC converters; working principles, modulation, design, and closed loop control of power factor correction converters (PFC): Boost type PFC, Totem-pole PFC, active front-end converter, three-phase PFCs; working principles, modulation, design, and closed loop control of single-stage AC-DC converters; G2V, V2X operations.

UNIT-III ***(10 hours)***

DC-DC Converter: Types of DC-DC converter used for EV chargers; working principles, modulation, design, modelling and closed loop control of dual active bridge, LLC converter, high frequency magnetics, soft-switching criteria.

UNIT-IV ***(13 hours)***

Protocols and communication: Open charge point protocol (OCPP), Open System Interconnection-Layer-Model (OSI), adapted PWM signal based low-level communication, PLC based high level communication, CAN communication, testing methodology for EV battery chargers and EVSE.

EMI/EMC considerations: sources of EMI, differential mode noise, common mode noise, LISN, measuring of EMI/EMC spectrum, design of DM filters, CM filters.

Case study: deployment of Bharat DC-001 and AC-001 chargers in Indian charging ecosystem.

Essential Readings:

1. Iqbal Husain, Electric and Hybrid Vehicles: Design Fundamentals, CRC Press. 2021.
2. L. Umanand, Power Electronics: Essentials and Applications, Wiley India, 2012.
3. N. Mohan, T.M. Underland and W.P. Robbins, Power Electronics – Converters, Applications and Design, 3rd Ed., Wiley India, 2008.

Suggested Readings:

1. Tom Denton, Automotive Electrical and Electronic Systems, Routledge, 5th Edition, 2018.
2. Robert W. Erickson, and Dragan Maksimovic, Fundamentals of Power Electronics, 3rd Ed., Springer, 2020.
3. Christoph Marscholik and Peter Subke, Road Vehicles - Diagnostic Communication, University Science Press, 2009.
4. Wolfhard Lawrenz, CAN System Engineering: From Theory to Practical Applications, Springer, 2013.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Study of EV Charging Infrastructure and Standards.
2. Analysis of Impact of EV Charging Load on Distribution System.
3. Simulation of Single-Phase AC–DC Rectifier with and without Power Factor Correction.
4. Design and Simulation of Single-Phase Boost PFC Converter for EV Charger.
5. Simulation and Performance Evaluation of Totem-Pole PFC Converter.
6. Analysis of Bidirectional AC–DC Converter for G2V and V2X Operation.
7. Design and Simulation of Dual Active Bridge (DAB) DC–DC Converter.
8. Design and Analysis of LLC Resonant DC–DC Converter for EV Charger.
9. High-Frequency Transformer Design for Isolated EV Charger.
10. Study and Implementation of EV Charger Communication Protocols.
11. EMI/EMC Analysis and Filter Design for EV Chargers.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Electrical Vehicle System Design (DSE-10/ GE-10)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electrical Vehicle System Design	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Objectives:

1. To understand vehicle fundamentals and dynamics relevant to electric vehicles.
2. To analyze and design EV powertrain systems and configurations.
3. To model, simulate, and size EV subsystems for performance and efficiency.
4. To integrate mechanical, electrical, and control aspects of EV design.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Analyze vehicle forces and performance parameters.
2. Compare conventional and electric vehicle powertrain configurations.
3. Design and size EV powertrain components.
4. Model EV systems and evaluate range and energy consumption.
5. Understand chassis, braking, steering, suspension, and their control in EVs.

UNIT-I ***(10 hours)***

Vehicle Fundamentals and Dynamics: Vehicle movement, rolling resistance, aerodynamic drag, grading resistance, tractive effort and power requirements, vehicle performance, acceleration characteristics, maximum speed and gradeability, slip ratio and traction limits, calculation of normal tire forces, effective tire radius and load transfer, static forces, longitudinal and cornering forces, interaction between longitudinal and lateral forces.

UNIT-II ***(12 hours)***

Conventional Vehicle Systems and Powertrain Fundamentals: Overview of conventional vehicle systems, IC engine components, operation of four-stroke engines, engine performance, supercharging and turbocharging techniques, combustion in spark ignition engines, engine emissions and emission control methods, automotive powertrain components; clutch, gearbox, torque converter, powertrain analysis, powertrain configurations; rear-wheel drive (RWD), front-wheel drive (FWD), and all-wheel drive (AWD).

UNIT-III ***(11 hours)***

Electric Vehicle Systems and Powertrain Design: Electric vehicle architecture and classifications, EV powertrain configurations and components, traction motors: DC, Induction, PMSM, SRM – characteristics and selection, tractive effort calculation, drive cycles, range modeling of battery electric vehicle, vehicle control unit (VCU) and system integration, auxiliary systems in EV, powertrain component sizing: motor, battery, inverter, Auxiliary control functions: start-stop, anti-roll, traction control.

UNIT-IV ***(12 hours)***

Chassis and vehicle structure: Vehicle and body center of gravity, mass moments of inertia, chassis stiffness, strength, and vibration behavior, external loads and structural considerations, multi-body vehicle models, ride comfort and NVH.

Steering, braking, and suspension: Steering systems: manual and power steering, brake systems: hydraulic, air brakes, braking analysis and weight transfer, Antilock Braking System (ABS), regenerative braking, suspension systems: dependent and independent.

Vehicle control systems: Steering control, braking control and electronic brake distribution, vehicle stability control, brake assist system, anti-spin regulation, suspension control: trim, damping, and roll control.

Essential Readings:

1. Mehrdad Ehsani et al., Modern electric, hybrid electric, and fuel cell vehicles, CRC press, 2018.
2. Iqbal Husain, Electric and hybrid vehicles: design fundamentals, CRC press, 2010.
3. Reza N. Jazar, Vehicle dynamics: theory and application, Springer, 2017.

Suggested Readings:

1. G. Genta, L. Morello, The Automotive Chassis, Volume 1: Components Design, Springer Nature, 2009.
2. Rajesh Rajamani, Vehicle Dynamics and Control, Springer, 2012.

Practical Component: ***(30 Hours)***

List of Experiments:

1. Calculation of road load forces and tractive effort using MATLAB.
2. Vehicle acceleration and gradeability analysis.
3. Modeling of IC engine torque-speed characteristics.
4. EV motor selection based on drive cycle requirements.
5. Battery pack sizing and range estimation for a BEV.
6. Simulation of regenerative braking and energy recovery.
7. EV powertrain modeling in MATLAB/Simulink.
8. Drive cycle-based energy consumption analysis.
9. Vehicle dynamics simulation for braking and steering.
10. Comparative study of ICE vs EV powertrain performance.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)