Summary of Minor Modifications in the MCA program syllabus

Some minor changes are proposed in the MCA program syllabus. These are listed below:

1. There are some intra-semester movements of some courses and minor updates in the names and contents of certain courses, listed below:

S.No.	Existing	Proposed	Change
1.	MCAC 102: Computer System Architecture (Semester I)	MCAC 102: Computer Organization and Architecture (Semester I)	Enhanced focus on organization aspect
2.	MCAC 203: Operating Systems (Semester II)	MCAC 105: Operating Systems (Semester I)	Moved from Semester II to Semester I
3.	MCAC 105: Data Mining (Semester I)	MCAC 203: Data Mining (Semester II)	Moved from Semester I to Semester II
4.	MCAC 204: Machine Learning (Semester II)	MCAC 204: Artificial Intelligence and Machine Learning (Semester II)	The change reflects an integrated approach to Artificial Intelligence and Machine Learning
5.	MCAE 309: Automata Theory (Semester III)	MCAE 203: Automata Theory (Semester II)	Moved from Semester III to Semester II

2. Core courses on (i) Statistical Methods and (ii) Artificial Neural Networks and Deep Learning have been introduced. Further, two credit courses on (i) Software Tools and Techniques and (ii) Reading Skills have been introduced.

3. Based on student feedback, elective courses on information retrieval, social networks, cloud computing, and data analysis and visualization have been introduced.

   Each theory course is a four-credit course and each practical course is a two-credit course. To pass a course, a student should obtain ‘D’ grade or a higher grade.

UNIVERSITY OF DELHI
MASTER OF COMPUTER APPLICATIONS (MCA)

I. MCA Programme Details:

Programme Objectives (POs):

Master of Computer Applications (MCA) is a full-time, four-semester course that includes one semester of project work in the fourth semester. The objective of the MCA programme is to impart quality education in computer science and its applications so that students are well prepared to face the challenges of the highly competitive IT industry. The course structure ensures the overall development of the student while concentrating on imparting the technical skills required for an IT profession. No wonder, today, after forty years of its existence, its alumni are holding important positions in the IT industry and academia in India and abroad.

Programme Specific Outcomes (PSOs):

The programme is designed to:

PSO1: enable the students to apply the computing and soft skills acquired in the MCA program to design and develop innovative applications for the betterment of society.
PSO2: provide exposure to techniques that would enable the students to design, implement, and evaluate IT solutions.
PSO3: To enable the students to meet the challenges of computer science and applications.

Programme Structure:

The MCA programme is a two-year course divided into four semesters. A student is required to complete 80 credits for the completion of the course and the award of a degree.

		Semester	***Semester***
Part – I	First Year	Semester I	Semester II
Part – II	Second Year	Semester III	Semester IV

Course Credit Summary

Semester	**Core Courses**			**Elective Course**			**Open Elective Course**			**Total Credits**
	No.of Courses	Credits (Th+T+P)	**Total Credits**	No. of Courses	Credits (Th+T+P)	**Total Credits**	No.of Courses	Credits (Th+T+P)	**Total Credits**	
I	6	16+0+6	**22**	0	0	**0**	0	0	**0**	**22**
II	5	12+0+6	**18**	1	3+0+1	**4**	0	0	**0**	**22**
III	3	9+0+3	**12**	1	3+0+1	**4**	1	3+0+1	**4**	**20**
IV	Project	0	**20**	0	0	**0**	0	0	**0**	**20**
Total Credits for the Courses			**72**			**8**			**4**	**84**

- English shall be the medium of instruction and examination.
- Examinations shall be conducted at the end of each semester as per the academic calendar notified by the University.
- The students will choose the elective courses out of the list of courses that are offered in a semester. An elective course offered by another department/ center/ institute may be taken subject to approval of the department.
- **Project:** Each student shall carry out a project in the fourth semester. The projects will be carried out under the supervision of the teacher(s) of the department. When the project is carried out in an external organization (academic institution/ industry), a supervisor may also be appointed from the external organization. The project work will be evaluated jointly by the internal supervisor and an examiner to be appointed by the department in consultation with the internal supervisor. The project evaluation shall be as follows:

  (a) Mid-semester evaluation: 30% weight
  (b) End-semester evaluation
    (i) Dissertation: 30% weight
    (ii) Viva-voce: 40% weight
- Examinations for courses specified in the odd (even) semesters shall be conducted only in the respective odd (even) semesters.
- **Promotion Criteria:** To be eligible for promotion to second year, a student must successfully complete at least 30 credits out of the courses prescribed for semester I and semester II, taken together.
- **Evaluation Criteria:** The scheme of evaluation shall be as follows: the performance of the students will be evaluated based on a comprehensive system of continuous and end-semester evaluation. For each course, there shall be one minor test, assignments/ laboratory work, quizzes, and an end-semester examination: (Mid-Term exam, assignments/practical & laboratory work - 30% weight; end-semester examination - 70% weight), except for practical courses where internal assessment and end-semester examination shall carry 50% weight each. The evaluation of the practical courses will be based on internal assessment and the end-semester evaluation by a board of examiners appointed by the Committee of Courses.
- In order to pass a course and earn credits prescribed for it, a student must obtain 'D' grade or a higher grade.
- **Conversion of Marks into Grades:**

Letter Grade	**Numerical Grade**	**Formula**	**Computation of grade cut off**
O (outstanding)	10	$m \geq \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.5\sigma$ or 90% whichever is lower
A+ (Excellent)	9	$\overline{X} + 2.0\sigma \leq m < \overline{X} + 2.5\sigma$	the value of $\overline{X} + 2.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 2.0\sigma$ or 80% whichever is lower
A (Very Good)	8	$\overline{X} + 1.5\sigma \leq m < \overline{X} + 2.0\sigma$	the value of $\overline{X} + 1.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.5\sigma$ or 70% whichever is lower
B+ (Good)	7	$\overline{X} + 1.0\sigma \leq m < \overline{X} + 1.5\sigma$	the value of $\overline{X} + 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} + 1.0\sigma$ or 60% whichever is lower
B (Above Average)	6	$\overline{X} \leq m < \overline{X} + 1.0\sigma$	the value of $\overline{X}$ to be taken into account for grade computation will be actual $\overline{X}$ or 50% whichever is lower

C (Average)	5	$\overline{X} - 0.5\sigma \leq m < \overline{X}$	the value of $\overline{X} - 0.5\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 0.5\sigma$ or 45% whichever is lower
D (Pass)	4	$\overline{X} - 1.0\sigma \leq m < \overline{X} - 0.5\sigma$	the value of $\overline{X} - 1.0\sigma$ to be taken into account for grade computation will be actual $\overline{X} - 1.0\sigma$ or 40% whichever is lower
F (Fail)	0	$\overline{X} - 1.0\sigma > m$	

- **CGPA to Percentage Conversion:**
  The formula for calculating the final percentage of marks from Cumulative Grade Point Average (CGPA) will be as per the University rules.
- **Eligibility for Award of Degree and Division Criteria:**
  A student would be eligible for the award of an MCA degree, provided he/ she earns the required number of credits. Such a student shall be categorized (on the basis of the CGPA to percentage conversion as per university rules) on the basis of the CGPA acquired during Part-I and Part-II examinations taken together, as follows:

  a) I Division: 60% or more marks in the aggregate
  b) II Division: 50% or more marks but less than 60% marks in the aggregate.
  c) Pass: 40% or more marks but less than 50% marks in the aggregate.

- **Attendance Requirement**

  No candidate shall be considered to have pursued a regular course of study unless he/she has attended 66.67% of the total number of classroom/ tutorial/ lab sessions conducted in each semester during his/her course of study. A student not complying with this requirement shall not be allowed to appear in the semester examinations. However, considering the merit of the case, the Head of the Department may condone the required percentage of attendance by not more than 10 percent during a semester.

- **Span Period**
  The span period will be four years from the date of registration in the programme.

- **Eligibility, Mode of Admissions, and Number of Seats in the MCA programme:**
  To be decided by the University in every academic year.

<u>Semester wise Details of MCA Course</u>

Semester I					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC101	Object Oriented Programming	3	0	1	**4**
MCAC102	Computer Organization and Architecture	3	0	1	**4**
MCAC103	Data Structures	3	0	1	**4**

MCAC104	Database Systems	3	0	1	**4**
MCAC105	Operating Systems	3	0	1	**4**
MCAC106	Software Tools and Techniques	1	0	1	**2**
	Total credits in core course	22			
	Number of elective courses	0			
	Total credits in elective course	0			
	Number of open electives	0			
	Total credits in elective course	0			
	Total credits in Semester I	**22**			

Semester II					
	Number of core courses	5			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC201	Design and Analysis of Algorithms	3	0	1	**4**
MCAC202	Data Communication and Computer Networks	3	0	1	**4**
MCAC203	Data Mining	3	0	1	**4**
MCAC204	Artificial Intelligence and Machine Learning	3	0	1	**4**
MCAC205	Reading Skills	0	0	2	**2**
	Total credits in core course	18			
	Number of elective courses	1			
	Credits in each elective	Theory	Tutorial	Practical	**Total**
	Elective course 1	3	0	1	**4**
	Total credits in elective courses	4			
	Number of open electives	0			
	Credits in each open elective	0			
	Total credits in open elective	0			
	Total credits in Semester II	22			

List of Elective Courses for Semester II		
Course Code	**Course Title**	**Th-T-P**
MCAE 201	Web Technologies	3-0-1
MCAE 202	Java Programming	3-0-1
MCAE 203	Automata Theory	3-0-1
MCAE 204	Statistical Methods	3-0-1

Semester III					
	Number of core courses	3			
Course Code	**Course Title**	**Credits in each core course**			
		Theory	Tutorial	Practical	**Total**
MCAC301	Information Security	3	0	1	**4**
MCAC302	Software Engineering	3	0	1	**4**
MCAC303	Artificial Neural Networks and Deep Learning	3	0	1	**4**
	Total credits in core course	**12**			
	Number of elective courses	1			
	Credits in each elective	Theory	Tutorial	Practical	**Total**
	Elective course I	3	0	1	**4**
	Total credits in elective courses	**4**			
	Number of open elective	1			
	Credits in each open elective	Theory	Tutorial	Practical	**Total**
	Open Elective I	3	0	1	**4**
	Total credits in open elective courses	**0**			
	Total credits in Semester III	**20**			

List of Elective Courses for Semester III		
Course Code	**Course Title**	**Th-T-P**
MCAE301	Compiler Design	3-0-1
MCAE302	Network Science	3-0-1
MCAE303	Natural Language Processing	3-0-1
MCAE304	Information Retrieval	3-0-1
MCAE305	Combinatorial Optimization	3-0-1
MCAE306	Digital Watermarking and Steganography	3-0-1
MCAE307	Quantum Computing	3-0-1
MCAE308	Computer Graphics	3-0-1
MCAE309	Digital Image Processing	3-0-1
MCAE310	Social Networks	3-0-1
MCAE311	Cloud Computing	3-0-1

List of Open Elective Courses for Semester III		
Course Code	**Course Title**	**Th-T-P**
MCAO301	Data Mining for Business Applications (for other departments)	3-0-1
MCAO302	Data Science using Python	3-0-1
MCAO303	Data Analysis and Visualization	3-0-1

Semester IV		
	Number of core courses	1
Course Code	**Course Title**	
MCAC401	Project work	20

	Number of elective courses	0
	Total credits in elective courses	0

	Number of open electives	0
	Total credits in open elective	0
	Total credits in Semester VI	**20**

Total Credits = 22 +22 + 20 + 20 = 84

Selection of Elective Courses:

The students may select the elective courses out of the list of courses that are offered in a semester.

II. Content Details for MCA Programme

SEMESTER – I

MCAC101: OBJECT ORIENTED PROGRAMMING [3-0-1]

Course Objectives: The course aims to develop the student's problem-solving skills. The course also focuses on debugging skills. The student learns to develop modular well documented code.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: select a suitable programming construct and in-built data structure for a given problem.

CO2: design, develop, document, and debug modular programs.

CO3: use recursion as a programming paradigm for problem-solving.

CO4: apply object-oriented paradigm for problem-solving.

Syllabus:

Unit-I Introduction: Notion of class, object, identifier, keyword, and literal; basic data types: int, float, string, Boolean; basic operators (arithmetic, relational, logical, assignment), standard libraries.

Unit-II Program Development: Modular program development, input and output statements, control statements: branching, looping, exit function, break, continue, and switch-break; arrays and pointers, testing and debugging a program.

Unit-III Recursion: Use of recursion as a programming paradigm for problem-solving.

Unit-IV Object Oriented Programming Concepts: Use of classes, inheritance, and Polymorphism

Unit-V Exception Handling and File Handling: Reading and writing text and structured files, errors and exceptions.

Readings:

1. R. G. Dromey, **How to Solve it by Computer,** Pearson, 2006.
2. Stanley B. Lippman, and Josée Lajoie, C++ PRIMER, Addison-Wesley, 2019.
3. Bjarne Stroustrup, **The C++ Programming Language (4th Edition)** Addison-Wesley, 2013

MCAC102: COMPUTER ORGANIZATION AND ARCHITECTURE [3-0-1]

Course Objectives: The course is designed to introduce basic building blocks of digital electronics, design and architecture of computer systems. This course aims to develop the skill to architect a digital computer system using a simulator.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the basic organization of computer hardware.

CO2: represent and manipulate data – number systems, conversion between different number systems, perform binary arithmetic.

CO3: design simple combinational and sequential logic circuits - flip-flops, counters, shift registers, adders, subtractor, multiplexer, demultiplexer, and Arithmetic/Logic unit.

CO4: design a CPU simple computer / microprocessor: instruction format, instruction set, addressing modes, bus structure, input/output architecture, memory unit, Arithmetic/Logic and control unit, data, instruction and address flow.

Syllabus:

Unit-I Basic Building Blocks: Boolean logic and Boolean algebra, tri-state logic; flip-flops, counters, shift registers, adders, subtractors, encoders, decoders, multiplexers, demultiplexers.

Unit-II Processor Design: CPU organization, register organization, stack organization, microprogrammed control unit, RISC architecture; microprocessor architecture, modern computing architectures. Bus and memory transfers, arithmetic, logic shift micro-operations; basic computer organization: common bus system, instruction formats, instruction cycle, interrupt cycle, input/output configuration.

Unit-III Memory Unit: Primary memory, secondary memory, associative memory, sequential access, direct access storage devices.

Unit-IV Input-Output Architecture: Input/Output devices; data transfer schemes - programmed I/O and DMA transfer; data transfer schemes for microprocessors.

Readings:

1. M. Morris Mano, **Computer System Architecture,** Revised 3rd Edition, Pearson, 2018.
2. W. Stallings, **Computer Organization and Architecture: Designing for Performance**, 9th Edition, Pearson Education, 2012.
3. A.S. Tanenbaum**, Structured Computer Organization**, 6th Edition, Prentice-Hall of India, 2012.
4. J.P. Hayes, **Computer System Architecture and Organization**, 3rd Edition, McGraw-Hill Education, 2017.

MCAC103: DATA STRUCTURES [3-0-1]

Course Objectives: The course is designed to develop the student's ability to develop algorithms for creating and manipulating basic data structures. The students also learn to compare and contrast different data structures for a given problem.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: develop programs using basic data structures: sets, lists, stacks, queues, trees, graphs and advanced data structures like balanced trees.

CO2: identify the suitable data structure for the problem at hand.

CO3: evaluate computational complexity of an algorithm.

Syllabus:

Unit-I Computational Complexity: Growth of functions, Recurrence relations: Asymptotic notations, solving recurrences using recursion trees.

Unit-II Basic Data Structures: Primitive data types, abstract data types, linear vs non linear data structures, arrays - static and dynamic, linked lists - single, doubly-linked, circular; stacks and queues using arrays and linked lists; operations, their analysis and applications.

Unit-III Trees: Binary tree, tree traversals, binary search tree, height balanced trees: AVL, red-black trees, B and B+ Trees, heaps, priority queues, operations, their analysis and applications.

Unit-IV Sets: Sets, multisets, maps, hash tables, dictionaries.

Unit-V Graphs: Representation of graphs, searching in graphs: breadth first search and its applications, depth first search and its applications, shortest paths, spanning trees.

Readings:

1. M. Goodrich, R. Tamassia, D. Mount, **Data Structures and Algorithms in C++/Java**, 2nd Edition, 2016, Wiley.
2. Elliot B. Koffman, Paul A.T. Wolfgang, **Objects, Abstraction, Data Structures and Design Using C++/Java**, 1st Edition, 2005, Wiley Global Education.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, **Introduction to Algorithms**, 3rd Edition, 2010, Prentice-Hall of India Learning Pvt. Ltd.

MCAC104: DATABASE SYSTEMS [3-0-1]

Course Objectives: This course aims to achieve competency in designing and developing databases for different applications, keeping in mind their constraints. The students learn to apply database design techniques to design a database and answer the user's queries.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: apply logical database design principles, including data normalization and E-R/EE-R diagrams to design an application.

CO3: choose suitable integrity constraints for an application.

CO4: answer database queries using Structured Query Language (SQL).

CO5: enumerate the concurrency control issues and discuss their resolution strategies.

CO6: design and implement database projects.

Syllabus:

Unit-I Introduction: Data modelling for a database, abstraction and data integration, three-level DBMS architecture.

Unit-II Database Design: Entity-Relationship model, Extended Entity-Relationship model.

Unit-III Relational Model and Relational Data Manipulations: Relation, conversion of ER diagrams to relations, integrity constraints, relational algebra, relational domain and tuple calculus.

Unit-IV Structured Query Language: DDL, DML, Views, Embedded SQL.

Unit-V Relational Database Design Concepts: Functional dependencies, determining keys, normalization-, lossless join and dependency preserving decomposition.

Unit-VI Transaction Management: ACID properties, concurrency control in databases, transaction recovery.

Unit-VII Introduction to NoSQL and XML Databases.

Readings:

1. A. Silberschatz, H. Korth and S. Sudarshan**, Database System Concepts,** 6th Edition**,** McGraw Hill, 2014.
2. R. Ramakrishnan and J. Gehrke, **Database Management Systems,** 3rd Edition, McGraw Hill, 2014.
3. Philip Lewis, Arthur Berstein and Michael Kifer, **Databases and Transaction Processing - An application-oriented Approach**, Prentice Hall, 2003

MCAC105: OPERATING SYSTEMS [3-0-1]

Course Objectives: This course aims to provide a functional perspective of the operating systems and develop skills to experiment with different components, tasks and services, including job scheduling, memory management, device management, process management and file management.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the basic functions of an Operating System.

CO2: distinguish between different types of operating systems to use each of them most efficiently in the respective application areas.

CO3: describe different techniques for managing computer resources like CPU, memory, file and devices.

CO4: implement algorithms for managing computer resources.

CO5: experiment with different components, tasks and services of the operating system.

Syllabus:

Unit 1- Introduction: operating system design goals, evolutionary history of operating systems; concept of user, job and resources; batch processing, multi-programming, time sharing; structure and functions of operating system.

Unit 2- Process Management: Process states, state transitions, process control structure, context switching, process scheduling, threads.

Unit 3- Memory Management: Address binding, dynamic loading and linking concepts, logical and physical addresses, contiguous allocation, fragmentation, paging, segmentation, combined systems, virtual memory, demand paging, page fault, page replacement algorithms, global vs local allocation, thrashing, working set model, pre-paging.

Unit 4- Process Synchronization: Process interaction, shared data and critical section, mutual exclusion, busy form of waiting, lock and unlock primitives, synchronization, classical problems of synchronization, semaphores, monitors, conditional critical regions, system deadlock, wait for graph, deadlock handling techniques: prevention, avoidance, detection and recovery.

Unit 5- File and Secondary Storage Management: File attributes, file types, file access methods, directory structure, file system organization and mounting, allocation methods, free space management; disk structure, logical and physical view, disk head scheduling, formatting, swap management.

Readings:

1. G. Gagne, A. Silberschatz, P. B. Galvin, and **Operating System Concepts** , 10th Edition,Wiley, 2018.
2. Gary Nutt, Nabendu Chaki, Sarmistha Neogy, **Operating Systems: A Modern Approach,** 3rd Edition, Addison Wesley, 2009.

3. D.M. Dhamdhere, **Operating Systems: A Concept Based Approach,** 2nd Edition, Tata McGraw-Hill, 2007.

MCAC106: SOFTWARE TOOLS AND TECHNIQUES [1-0-1]

Course Objective:

To develop proficiency in the use of software tools required for project development.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: use the command line interface efficiently
CO2: use features of version control systems
CO3: debug and profile code
CO4: manage dependencies

Syllabus:

Shell tools and scripting, editors (Vim), data wrangling, command-line environment, version control (Git), debugging and profiling, metaprogramming: working with daemons, FUSE, backups, APIs, common command-line flags/patterns, window managers, VPNs, Markdown, Booting + Live USBs, Docker, Vagrant, VMs, cloud, OpenStack, notebook programming

Readings:

1. C. Newham, Learning the Bash Shell: **Unix shell programming**. O'Reilly Media, Inc.; 2005.
2. W. Shotts, **The Linux command line: a complete introduction**. No Starch Press; 2019.
3. https://git-scm.com/book/en/v2

SEMESTER – II

MCAC201: DESIGN AND ANALYSIS OF ALGORITHMS [3-0-1]

Course Objectives: The course introduces techniques for algorithm design and analysis of the asymptotic performance of these algorithms. This course aims to achieve competence in designing efficient algorithms using different data structures for real-world problems.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe various algorithm design techniques, including iteration, divide and conquer, dynamic programming, and greedy approach algorithms.

CO2: analyze the strengths and weaknesses of each technique.

CO3: identify and apply technique(s) suitable for simple applications.

CO4: demonstrate the correctness of algorithms and analyze their time complexity.

Syllabus:

Unit-I Computational Complexity: Review of growth of functions, asymptotic notation, Master's theorem.

Unit-II Iterative Algorithms: Searching and sorting techniques - linear search, binary search, insertion sort – time complexity and proof of correctness.

Unit-III Divide and Conquer: Binary search, merge sort and quick sort – time complexity

Unit-IV Lower bounding techniques: Decision Trees.

Unit-V Linear Sorting: Count sort, radix sort, bucket sort.

Unit-VI Greedy Algorithms: Interval scheduling, minimum spanning trees – Prim's algorithm, Kruskal algorithm, shortest path problem – Djikstra's algorithm.

Unit-VII Dynamic Programming: Weighted interval scheduling, matrix chain multiplication, knapsack problem, longest common subsequence.

Unit-VIII String Processing: Brute-force method, KMP algorithm.

Unit-IX Randomized algorithms: Introduction to random numbers, randomized Qsort, randomly built BST.

Readings:

1. J. Kleinberg and E.Tardos, **Algorithm Design**, 1st Edition, Pearson Education India, 2013.
2. T. H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, **Introduction to Algorithms**, 3rd Edition, Prentice-Hall of India, 2010.
3. Sara Baase, Allen Van Gelder, **Computer Algorithm – Introduction to Design and Analysis**, 3rd edition, Pearson Education, 2002.
4. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, **Algorithms**, 1st Edition, Tata McGraw Hill, 2017.
5. Richard Johnsonbaugh and Marcus Schaefer, **Algorithms**, Pearson Education India, 2014.

MCAC202: DATA COMMUNICATION AND COMPUTER NETWORKS [3-0-1]

Course Objectives: The course introduces the basic concepts of data communication techniques, and computer network, various protocols and their applications. It also aims to develop skills for interworking between computer networks and switching components in telecommunication systems.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply data communication techniques in real-life experiments like telemetry and develop skills to modify the existing ones to better suit in different situations.

CO2: develop skills to apply services of various computer networks in various technical and professional fields.

CO3: reduce the overheads of different reference models and optimize their performances.

CO4: develop skills to apply, modify and develop new protocols in different layers of existing protocol stacks to suit customized requirements.

CO5: use various network applications to avail network services efficiently and develop skills to design new applications to open new services.

Syllabus:

Unit-I Data Communication Techniques: Theoretical basis of data communication, analog and digital signals, time domain and frequency domain analysis, frequency spectrum and bandwidth, asynchronous and synchronous transmission, data encoding and modulation techniques, baseband and broadband transmission, pulse code modulation, baud rate and bitrate of a channel, multiplexing- FDM and TDM, transmission medium, transmission errors – error detection techniques.

Unit-II Network Classification and Network Services: Local Area Networks, Metropolitan

Area Networks, Wide Area Network, wireless networks, internetworking and Internet, business and home applications, mobile user services.

Unit-III Network Architecture and Reference Models: Layered network architectures, protocol hierarchies, interface and services, ISO-OSI reference model, TCP/IP reference model, Internet protocol stack.

Unit-IV Data Link Layer Functions and Protocols: Framing, flow-control, error recovery protocols, Data link layer of internet-PPP protocol.

Unit-V Medium Access Sublayer: CSMA/CD protocol and Ethernet, hubs and switches, fast Ethernet, gigabit Ethernet, CSMA/CA protocol and WLAN.

Unit-VI Network and transport layers functions and protocols: Network switching mechanisms- circuit switching, packet switching, routing and congestion control, TCP/IP protocol architecture.

Unit-VII Network Applications: File transfer protocol, electronic mail, World Wide Web.

Readings:

1. A S Tanenbaum, **Computer Networks**, 5th Edition, Pearson Education India, 2013
2. Behrouz A Forouzan, **Data Communications and Networking**, 5th Edition, McGraw Hill Education, 2017.

MCAC203: DATA MINING [3-0-1]

Course Objectives: This course introduces the KDD process. It would enable students to translate real-world problems into predictive and descriptive tasks. The course also covers data cleaning and visualization, as well as supervised and unsupervised mining techniques.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: experiment with the basic data exploration methods to develop an understanding of given data.

CO2: identify a suitable pre-processing method for a given problem.

CO3: implement and apply appropriate data mining algorithms for a given problem.

CO4: use GUI-based mining softwares.

Syllabus:

Unit-I Overview: The process of knowledge discovery in databases, predictive and descriptive data mining techniques, supervised and unsupervised learning techniques.

Unit-II Data preprocessing: Data cleaning, data transformation, data reduction, discretization.

Unit-III Classification: Curse of dimensionality, overfitting and underfitting, decision trees, decision rules, Naïve Bayes classifier, instance-based methods, model evaluation and validation methods.

Unit-IV Clustering: Basic issues in clustering, k-means clustering, hierarchical clustering, density-based methods, cluster validation methods and metrics.

Unit-V Association Rule Mining: Frequent item sets, closed and maximal item sets, Apriori algorithm for association rule mining.

Readings:

1. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2016.
2. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques,** 3rd Edition, Morgan Kaufmann, 2011.
3. Charu C Agrawal, **Data Mining: The Textbook**, Springer, 2015.

MCAC204: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING [3-0-1]

Course Objectives: Beginning with a comprehensive overview of the AI techniques, the course introduces the supervised and unsupervised machine learning (ML) techniques, alongwith their applications in solving real-world problems. The course also covers evaluation and validation methods for ML models.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: discuss Turing Test, and various methods of knowledge representation as applicable to a given context.

CO2: design and implement supervised and unsupervised machine learning algorithms for real-world applications while understanding the strengths and weaknesses.

CO3: analyse the computational complexity of various machine learning algorithms.

CO4: fine-tune machine learning algorithms and evaluate models generated from data.

Syllabus:

Unit-I Introduction to Artificial Intelligence: Evolution of artificial intelligence (AI) as a discipline, definitions and approaches, philosophical issues, AI for all, ethical issues and responsible AI.

Unit-II Introduction to Machine Learning: Hypothesis and target class, bias-variance tradeoff, Occam's razor, approximation and estimation errors, curse of dimensionality, dimensionality reduction, feature scaling, feature selection methods.

Unit-III Regression: Linear regression with one variable, linear regression with multiple variables, gradient descent, logistic regression, polynomial regression, over-fitting, regularization. performance evaluation metrics, validation methods.

Unit-IV Classification: Decision trees, Naive Bayes classifier, perceptron, multilayer perceptron, neural network, back-propagation algorithm, support vector machine, kernel functions.

Unit V Evaluation: Performance evaluation metrics, ROC Curves, validation methods, bias-variance decomposition, model complexity.

Unit-VI Unsupervised Learning: Clustering, distance metrics, mixture models, expectation maximization, cluster validation methods.

Readings:

1. E. Alpaydin, **Introduction to Machine Learning,** MIT press, 2014.
2. T. M. Mitchell, **Machine Learning,** McGraw Hill Education, 2017.
3. Christopher M. Bishop, **Pattern Recognition And Machine Learning**, Springer-Verlag, 2016.
4. Shai Shalev-Shwartz, Shai Ben-David, **Understanding Machine Learning: From Theory to Algorithms,** Cambridge Press, 2014.
5. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, eds. **Machine learning: An artificial intelligence approach**, Springer Science & Business Media, 2013.

MCAC205: READING SKILLS [0-0-2]

Course Objectives: The course aims to develop an important skills of independent reading.

Course Learning Outcomes:

On completing this course, a student will be able to:

CO1: Develop a habit of independent reading.

CO2: Given a requirement, independently select sources of reading.

CO3: Read and assimilate independently.

This is a self-study course. The students will carry out extensive reading on a topic to be assigned by the department.

MCAE201: WEB TECHNOLOGIES [3-0-1]

Course Objectives: The course aims to develop skills for static and dynamic web development, including the use of front-end designing tools and client-side and server-side programming. It also sensitizes and prepares the student to apply security and privacy concerns in an application.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe Internet, World Wide Web, client-server architecture and communication protocols.

CO2: design multi-platform web applications.

CO3: apply various client-side web technologies for front-end development.

CO4: work with server-side web technologies.

CO5: apply web technologies for retrieval of information.

Syllabus:

Unit-I Introduction: Introduction to networking, TCP/IP, DNS, Internet and its evolution, World Wide Web, Web 2.0, Web 3.0, network communication protocols (HTTP/HTTPS, SMTP, IMAP, POP, FTP), client-server architecture, web applications architecture, application and web servers, web clients.

Unit-II Front-end Development: Introduction to HTML5, HTML elements, HTML tags, lists, tables, frames, forms, basics of XHTML, CSS style sheets, DOM, XML, XSLT

Unit-III Client-Side Programming: JavaScript syntax, variables and data types, literals, functions, objects, arrays, built-in objects, event handling, modifying element style, document trees.

Unit-IV Server-Side Programming: Creation of dynamic content, server-side programming using Java Servlets, web services, session management, introduction to server-side scripting, accessing database from the front-end.

Unit-V Web Security, Cookies and Authentication: Security threats, security risks of a website, web attacks and their prevention, web security model, setting, accessing and destroying cookies, anonymous access, authentication by IP address and domain, Integrated Windows Authentication, digital signatures, digital certificates, firewalls.

Readings:

1. Jeffery C. Jackson, **Web Technologies: A Computer Science Perspective**, Pearson Education India, 2007.
2. Achyut Godbole and Atul Kahate, **Web Technologies: TCP/IP, Web/Java Programming, and Cloud Computing,** 3rd Edition, McGraw-Hill Education, 2013.
3. Roger S Pressman and David Lowe, **Web Engineering: A Practitioner's Approach,** Tata McGraw-Hill, 2017.
4. Mark Pilgrim, **HTML5: Up and Running**, O'Reilly | Google Press, 2010 .
5. Jim Keogh, **J2EE: The Complete Reference**, McGraw Hill Education, 2017.

MCAE202: JAVA PROGRAMMING [3-0-1]

Course Objectives: This course introduces object-oriented concepts through Java programming. The course also include reading and writing files using exception handling.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply the object-oriented concepts: Classes, Objects, Inheritance, Polymorphism, for problem solving.

CO2: handle program exceptions.

CO3: design, implement, document, test, and debug a Java application consisting of multiple classes.

CO4: handle input/output through files.

CO5: create Java applications with graphical user interface (GUI).

Syllabus:

Unit-I Introductory Concepts: program, identifiers, variables, constants, primitive data types, expressions, control statements, structured data types, arrays, functions.

Unit-II Object Oriented Concepts: Abstraction, encapsulation, objects, classes, methods, constructors, inheritance, polymorphism, static and dynamic binding, overloading, abstract classes, interfaces and packages.

Unit-III File Handling: Byte stream, character stream, file I/O basics, file operations, serialization.

Unit-IV Exception Handling: Throw and catch exceptions, throw, try, and catch blocks, multiple catch blocks, finally clause, throwable class, types of exceptions, java.lang exceptions, built-in exceptions.

Unit-V GUI Design: GUI based I/O, input and message dialog boxes, swing components, displaying text and images in window.

Readings:

1. James Gosling, Bill Joy, Guy L. Steele Jr, Gilad Bracha, Alex Buckley, **The Java Language Specification, Java SE,** 7th Edition, Addison-Wesley, 2013**.**
2. Cay S. Horstmann, **Core Java - Vol. I – Fundamentals,** 10th Edition, Pearson, 2017**.**
3. Deitel and Deitel, **Java-How to Program,** 9th Edition, Pearson Education, 2012.
4. Richard Johnson, A**n Introduction to Java Programming and Object-Oriented Application Development,** Thomson Learning, 2006.
5. Herbert Schildt, **Java: The Complete Reference,** 10th Edition, McGraw-Hill Education, 2018.

MCAE203: AUTOMATA THEORY [3-0-1]

Course Objectives: The course introduces the theoretical models of computation and their limitations. The students will develop skills to analyze automata and their computational power to recognize languages. The students will learn to apply the knowledge of automata theory, grammar, and Turing machines for solving problems in language translation.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: describe the mathematical model of machines.

CO2: examine whether a language is regular or context free.

CO3: identify ambiguity in grammar

CO4: construct a parse tree for the given grammar.

CO5: design finite automaton, push down automator, Turing machine for a language.

CO6: examine Turing decidability of a language.

Syllabus:

Unit-I Introduction: Alphabets, strings, and languages.

Unit-II Finite Automata and Regular Languages: Deterministic and non-deterministic finite automata, regular expressions, regular languages and their relationship with finite automata, pumping lemma and closure properties of regular languages.

Unit-III Context Free Grammars and Pushdown Automata: Context free grammars (CFG), parse trees, ambiguities in grammars and languages, pushdown automaton (PDA) and the language accepted by PDA, deterministic PDA, Non- deterministic PDA, properties of context free languages; normal forms, pumping lemma, closure properties, decision properties.

Unit-IV Turing Machines: Turing machine as a model of computation, programming with a Turing machine, variants of Turing machine and their equivalence.

Unit-V Undecidability: Recursively enumerable and recursive languages, undecidable problems about Turing machines: halting problem, Post Correspondence Problem, and undecidability problems about CFGs.

Readings:

1. J. E. Hopcroft, R. Motwani, and J. D. Ullman, **Introduction to Automata Theory, languages, and computation**, 2016.
2. H.R. Lewis, C.H. Papadimitriou, C. Papadimitriou, **Elements of the Theory of Computation**, 2nd Edition, Pearson Education, 2015
3. P. Linz, **Introduction to Automata Theory, Languages, and Computation**, Jones and Bartlett, 2016.

MCAE 204 Statistical Methods

Course Objectives: To equip students with the skills necessary to apply statistical methods for various applications.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: apply descriptive statistical techniques to summarize and interpret data

CO2: apply inferential statistical methods, including hypothesis testing and confidence interval estimation.

CO3: perform and interpret simple and multiple linear regression analysis

CO4: apply principles of experimental design in the context of a problem

Syllabus:

Unit-1 Introduction: Descriptive statistics: measures of central tendency and variability, representation of data: stem and leaf diagram, histogram, boxplot, and ogive; bar diagram and its variations, pie charts; probability distributions: discrete and continuous, joint and conditional probability; theory of attributes: coefficient of association and coefficient of colligation.

Unit-II: Statistical Inference: Parameter and statistic; sampling distributions, confidence intervals and margin of error, hypothesis testing; non-parametric inference: non-parametric tests: Mann-Whitney U test, Kruskal-Wallis test, Spearman's rank correlation coefficient.

Unit-III Regression and Classification: Correlation: measure and significance, simple linear regression, multiple linear regression, one-way classification, analysis of variance, two-way classification, analysis of covariance, curvilinear regression, factorial experiments.

Readings:

1. Robert S. Witte and John S. Witte, **Statistics**, John Wiley & Sons Inc; 11th edition, 2021
2. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, **An Introduction to Statistical Learning**, Springer, 2023.
3. G. W. Snedecor, W. G. Cochran, **Statistical Methods**, Iowa State University Press, 1973
4. John A. Rice, **Mathematical Statistics and Data Analysis**, Cengage, 2013

SEMESTER–III

MCAC301: INFORMATION SECURITY [3-0-1]

Course Objectives: The course aims to train the students to maintain the confidentiality, integrity and availability of data. The student learns various data encryption protocols for transmitting data over unsecured channels in a network.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe various security issues.

CO2: implement symmetric and asymmetric cryptographic methods.

CO3: describe the role and implementation of digital signatures.

CO4: make use of software tools for intrusion detection, auditing and logging.

Syllabus:

Unit-I Overview of Security: Protection versus security; aspects of security– confidentiality, data integrity, availability, privacy; user authentication, access controls, Orange Book Standard.

Unit-II Security Threats: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats: tapping and piracy.

Unit-III Cryptography: Substitution, transposition ciphers, symmetric-key algorithms: Data Encryption Standard, Advanced Encryption Standard, IDEA, block cipher operation, stream ciphers: RC-4. Public key encryption: RSA, ElGamal. Diffie-Hellman key exchange. Elliptic Curve, EC cryptography, Message Authentication code (MAC), cryptographic hash function.

Unit-IV Digital signatures: ElGamal digital signature scheme, Elliptic Curve digital signature scheme, NISTdigital signature scheme.

Unit-V Key Management and Distribution: Symmetric key distribution, X.509 Certificate public key infrastructures.

Unit-VI Cyber Security: Cyberspace, cyber crimes, cybercriminals, cyber security, cyber security threats, DOS attack, vulnerability analysis, firewall, intrusion detection system.

Readings:

1. W. Stalling, **Cryptography and Network Security Principles and Practices,** 7th Edition, Pearson Education of India, 2018.
2. A. J. Elbirt, **Understanding and Applying Cryptography and Data Security**, CRC Press, Taylor Francis Group, New York, 2015.
3. C. Pfleeger and S. L. Pfleeger, Jonathan Margulies, **Security in Computing** (5th Edition,, Prentice-Hall of India, 2015.
4. Peter W. Singer and Allan Friedman, **Cybersecurity and Cyberwar**, Oxford University Press, 2014.
5. Jonathan Clough, **Principles of Cybercrime**, Cambridge University Press, 2015.

MCAC302: SOFTWARE ENGINEERING [3-0-1]

Course Objectives: The course targets developing skills that would enable the students to construct high-quality software by applying the established software design practices for analysis, design, implementation, testing and maintenance.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the software engineering layered technology and compare software

process models.

CO2: apply agile development methods for developing software.

CO3: describe software/system requirements and the processes involved in the discovery and documentation of these requirements.

CO4: apply system modelling techniques and object-oriented design for software development.

CO5: test software using verification and validation, static analysis, reviews, inspections, and audits.

CO6: describe steps in software project management

CO7: work individually and in a team to develop and deliver quality software.

Syllabus:

Unit-I Software Engineering: The software crisis, software engineering principles, programming-in-the-small vs programming-in-the-large.

Unit-II Software process: The software lifecycle, the waterfall model and variations, risk-driven approaches, introduction to evolutionary and prototyping approaches, agile process models, and system classifications, Agile approach to software engineering.

Unit-III Project management: Relationship to lifecycle, project planning, project control, project organization, risk management, cost models, configuration management, version control, quality assurance, metrics.

Unit-IV Software requirements: Requirements analysis, functional and non-functional requirements elicitation, analysis tools, requirements definition, requirements specification, static and dynamic specifications, requirements review.

Unit-V Software design: Design for reuse, design for change, design notations, design evaluation and validation.

Unit-VI Implementation and Maintenance: Programming standards and procedures, modularity, data abstraction, static analysis, unit testing, integration testing, regression testing, verification and validation, tools for testing, fault tolerance, the maintenance problem, the nature of maintenance, planning for maintenance**.**

Readings:

1. R.S. Pressman, **Software Engineering: A Practitioner's Approach,** 7th Edition, McGraw-Hill, 2010.
2. I. Sommerville, **Software Engineering,** 10th Edition, Pearson Education, 2015.
3. R. Mall, **Fundamentals of Software Engineering,** 4th Edition, Prentice-Hall of India, 2014.
4. K.K. Aggarwal and Y. Singh, Software Engineering, 3rd Edition, New Age International Publishers, 2008.
5. P. Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2005.
6. N.S. Godbole, Software Quality Assurance: Principles and Practice for the new Paradigm, 2nd Edition, Alpha Science, 2007.

MCAC 303: ARTIFICIAL NEURAL NETWORKS AND DEEP LEARNING [3-0-1]

Course Objectives: The student learns various state-of-the-art deep learning algorithms and their applications to solve real-world problems. The student develops skills to design neural network architectures and training procedures using various deep learning platforms and software libraries.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: describe the feedforward and deep networks.

CO2:design single and multi-layer feed-forward deep networks and tune various hyper-parameters.
CO3: Use GANs to generate models that generate patterns.
CO4: Apply Large Language Models to an NLP task.
CO5: analyze the performance of deep networks.

Syllabus:

Unit-I Introduction: Historical context and motivation for deep learning; basic supervised classification task, optimizing logistic classifier using gradient descent, stochastic gradient descent, momentum, and adaptive subgradient method.

Unit-II Neural Networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyperparameter tuning.

Unit-III Convolution Neural Networks: Introduction to convolution neural networks: stacking, striding and pooling, applications like image, and text classification.

Unit-IV Sequence Modeling: Recurrent Nets: Unfolding computational graphs, recurrent neural networks (RNNs), bidirectional RNNs, encoder-decoder sequence to sequence architectures, deep recurrent networks.

Unit-V Autoencoders: Undercomplete autoencoders, regularized autoencoders, sparse autoencoders, denoising autoencoders, representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Unit VI: Generative Adversarial Networks (GANs): Introduction to Generative Adversarial Networks, GAN Architectures (DCGAN, CycleGAN), applications of GANs (Image Generation, Style Transfer)

Unit VII: Large Language Models: Introduction to Natural Language Processing (NLP), Traditional NLP Techniques, transformer architecture, pre-training and fine-tuning language models, ethical considerations and bias in language models, applications of large language models (text generation, sentiment analysis, question answering)

Unit-VIII Structuring Machine Learning Projects: Orthogonalization, evaluation metrics, train/dev/test distributions, size of the dev and test sets, cleaning up incorrectly labelled data, bias and variance with mismatched data distributions, transfer learning, multi-task learning.

Readings:

1. Ian Goodfellow, **Deep Learning**, MIT Press, 2016.
2. Jeff Heaton, **Deep Learning and Neural Networks**, Heaton Research Inc, 2015.
3. Mindy L Hall, Deep Learning, VDM Verlag, 2011.
4. Li Deng (Author), Dong Yu, **Deep Learning: Methods and Applications (Foundations and Trends in Signal Processing)**, Now Publishers Inc, 2009.

MCAE 301: COMPILER DESIGN [3-0-1]

Course Objectives: The course aims to develop the ability to design, develop, and test a functional compiler/ interpreter for a subset of a popular programming language.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe how different phases of a compiler work.

CO2: implement top-down and bottom-up parsing algorithms.

CO3: use tools like Lex and Yacc to implement syntax-directed translation.

Syllabus:

Unit- I Lexical and Syntactic Analysis: Review of regular languages, design of a lexical analyzer generator, context-free grammars, syntactic analysis: top-down parsing: recursive descent and predictive parsing, LL(k) parsing; bottom-up parsing: LR parsing, handling ambiguous in bottom-up parsers.

Unit-II Syntax directed translation: Top-down and bottom-up approaches, data types, mixed mode expression; subscripted variables, sequencing statement, subroutines and functions: parameters calling, subroutines with side effects.

Unit-III Code generation, machine dependent and machine-independent optimization techniques.

Readings:

1. A.V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, **Compilers, Principles, Techniques and Tools,** Pearson, 2016.

2. Dick Grune, Kees van Reeuwijk, Henri E .Bal, Ceriel J.H. Jacobs, K Langendoen, **Modern Compiler Design,** Springer, 2012.

MCAE302: NETWORK SCIENCE [3-0-1]

Course Objectives: The course aims to acquaint the students with the graph theory concepts relevant for network science. The students learn dynamics of networks in the context of applications from disciplines like biology, sociology, and economics.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: discuss the ubiquity of graph data model.

CO2: identify the structural features of a network

CO3: describe the graph generation models

CO4: identify community structures in networks

CO5: write programs to solve complex network problems

Syllabus:

Unit-I Introduction: Introduction to complex systems and networks, modeling of complex systems, review of graph theory.

Unit-II Network properties: Local and global properties like clustering coefficient, eccentricity; centrality measures for directed and undirected networks.

Unit-III Graph models: Random graph model, Small world network model, Barabasi-Albert (preferential attachment) network model.

Unit-IV Community structure in networks: Communities and community detection in networks, hierarchical algorithms for community detection, modularity based community detection algorithms, label propagation algorithm.

Readings:

1. Mohammed J. Zaki, Wagner Meira Jr.; **Data Mining and Analysis: Fundamental Concepts and Algorithms**, Cambridge University Press, 2014.

2. Albert Barabasi, **Network Science** , Cambridge University Press, 2016.

3. David Easley and Jon Kleinberg, **Networks, Crowds, and Markets: Reasoning About a Highly Connected World**, Cambridge University Press, 2010.

MCAE 303: NATURAL LANGUAGE PROCESSING [3-0-1]

Course Objectives: The course provides a rigorous introduction to the essential components of a Natural Language Processing (NLP) system. The students will learn various statistical, machine learning, and deep learning techniques in NLP and apply them to solve machine translation and conversation problems.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: compare and contrast varius language models.

CO2: compare and contrast various machine translation approaches.

CO3: compare and contrast various text summarization techniques.

CO4: implement an NLP system.

Syllabus:

UNIT I Introduction: Natural Language Processing (NLP), history of NLP, neural networks for NLP, applications: sentiment analysis, spam detection, resume mining, conversation modeling, chat-bots, dialog agents, question processing.

UNIT II Language Modeling and Part of Speech Tagging: Unigram language model, bigram, trigram, n-gram, advanced smoothing for language modeling, empirical comparison of smoothing techniques, applications of language modeling, natural language generation, parts of speech tagging, morphology, named entity recognition.

UNIT III Words and Word Forms: Bag of words, skip-gram, continuous bag-of-words, embedding representations for words lexical semantics, word sense disambiguation, knowledge based and supervised word sense disambiguation.

UNIT IV Text Analysis, Summarization and Extraction: Sentiment mining, text classification, text summarization, information extraction, named entity recognition, relation extraction, question answering in multilingual setting; NLP in information retrieval, cross-lingual IR

UNIT V Machine Translation: Need of machine translation, Problems of machine translation, mt approaches, direct machine translations, rule-based machine translation, knowledge based MT System, Statistical Machine Translation (SMT), parameter learning in SMT (IBM models) using EM, Encoder-decoder architecture, neural machine translation.

Readings:

1. Dan Jurafsky and James H. Martin, **Speech and Language Processing,** Pearson, 2009.
2. Jacob Eisenstein, **Introduction to Natural Language Processing,** MIT Press, 201 .
3. Yoav Goldberg, **Neural Network Methods for Natural Language Processing**. Morgan and Claypool Publisher (2017).
4. Jason Brownlee, **Deep Learning for Natural Language Processing,** Machine Learning Mastery, 2019.
5. Steven Bird, Ewan Klein and Edward Loper, **Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit**, O'Reilly, 2009.

MCAE 304: INFORMATION RETRIEVAL [3-0-1]

Course Objectives: This course aims to equip the students with basic techniques for information retrieval that find use in text analytics. The student will also learn to apply the tools for information extraction.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe early developments in IR.

CO2: apply measures for evaluating retrieved information.

CO3: choose appropriate model for document processing.

CO4: apply available tools for information retrieval.

CO5: develop simple information retrieval tools to solve real world problems.

Syllabus:

Unit 1- Introduction: Information, Information need and relevance; The IR system; early developments in IR, user interfaces.

Unit 2- Retrieval and IR Models: Boolean retrieval; term vocabulary and postings list; index construction; ranked and other alternative retrieval models.

Unit 3- Retrieval Evaluation: Notion of precision and recall; precision-recall curve, standard performance measures such as MAP, reciprocal ranks, F-measure, NDCG, rank correlation.

Unit 4- Document Processing: Representation; Vector space model; feature selection; stop words; stemming; notion of document similarity; standard datasets..

Unit 5- Classification and Clustering: Notion of supervised and unsupervised algorithms; naive bayes, nearest neighbour and rochio's algorithms for text classification; clustering methods such as k-means.

Unit-6: Link Analysis: Page Rank, HITs, web crawling. applications.

Readings:

1. R. Baeza-Yaets, B. Ribeiro-Neto, **Modern Information Retrieval: The Concept and Technology behind Search**, Latest Edition, Addison-Wesley, 1999.

2. C. D. Manning, P. Raghvan, H. Schutze, **Introduction to Information Retrieval,** Cambridge University Press, 2008.

3. D. A. Grossman, O. Frieder, **Information Retrieval: Algorithms and Heuristics,** 2nd Ed., Springer, 2004.

4. S. Buettcher, Charles L.A. Clarke, G. V. Carmack, **Information Retrieval: Implementing and Evaluating Search Engines,** MIT Press.

5. B. Croft, D. Metzler, T. Strohman, **Search Engines: Information Retrieval in Practice,** Addison Wesley

MCAE 305: COMBINATORIAL OPTIMIZATION [3-0-1]

Course Objectives: This course aims to introduce the mathematical tools for solving and analyzing combinatorial optimization problems. The course develops skills for the formulation of problems in different computing and data science domains as combinatorial optimization problems.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: differentiate between the computational complexities of linear programming (LP) and integer programming (IP).

CO2: construct and solve LP and IP problems for a given application.

CO3: apply polyhedral analysis to develop algorithms.

CO4: use the concept of duality to design exact and approximate algorithms.

Syllabus:

Unit-I Introduction: Optimization problems, neighborhoods, local and global optima, convex sets and functions, simplex method, degeneracy; duality and dual simplex algorithm, computational considerations for the simplex and dual simplex algorithms-Dantzig-Wolfe algorithms.

Unit-II Integer Linear Programming: Cutting plane algorithms, branch and bound technique and approximation algorithms for traveling salesman problem.

Unit-III Graph Algorithms: Primal-Dual algorithm and its application to shortest path (Dijkstra's algorithm, Floyd-Warshall algorithms), max-flow problem (Ford and Fulkerson labeling algorithms), matching problems (bipartite matching algorithm, non-bipartite matching algorithms, bipartite weighted matching-hungarian method for the assignment problem, non-bipartite weighted matching problem), efficient spanning tree algorithms.

Unit-IV Matroids: Independence systems and matroids, duality, matroid intersection.

Readings:

Bernhard Korte and Jens Vygen, **Combinatorial Optimization: Theory and Algorithms**, 6^{th} edition, Springer, 2018.

J. Matousek and B. Gartner, **Understanding and Using Linear Programming**, Springer, 2017..

C.H. Papadimitriou and K.Steiglitz, **Combinatorial Optimization: Algorithms and complexity**, Dover Publications, 1998.

Mokhtar S.Bazaraa, John J. Jarvis and Hanif D. Sherali, **Linear Programming and Network Flows**, 4^{th} Edition, Wiley-Blackwell, 2010.

1. H. A. Taha, **Operations Research: An Introduction**, 8^{th} edition, Pearson Education India, 2014.

MCAE 306: DIGITAL WATERMARKING AND STEGANOGRAPHY [3-0-1]

Course Objectives: The students will learn about the need and various approaches of information hiding, steganography, watermarking models and message coding, watermark security and authentication. The students will develop skills to implement these techniques in real-world scenarios.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: implement algorithms for steganography and watermarking operations.

CO2: evaluate steganography and watermarking algorithms for robustness.

CO3: describe various fingerprinting techniques.

Syllabus:

Unit-I Information hiding: Introduction, background, and applications of information hiding: data hiding, applications of data hiding.

Unit-II Steganography: Frameworks of secret communication, security of steganography systems, information hiding in noisy data, adaptive and non-adaptive algorithms, active and malicious attackers, information hiding in written text, invisible communication.

Unit-III Data hiding in still images: LSB encoding, BPCS steganography, lossless data hiding, data hiding by quantification, patchwork, transform domain methods, robust data hiding in JPEG images, frequency domain watermarking detecting malicious tempering, robust wavelet-based watermarking, Kundur-Hatzinakos watermarking, data hiding in binary images, Zhao-koch method, Wu-Lee method, CPT method, TP method, data hiding in fax images.

Unit-IV Watermarking: Introduction, watermarking principals, applications, requirements and algorithmic design issues, evaluation and standards of watermarking.

Unit-V Fingerprinting: introduction, terminology and requirements, classifications, fingerprinting schemes, statistical fingerprinting, and collusion-secure fingerprinting.

Readings:

1. I. J.Cox, M. L.Miller, J. A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Stegonagraphy, Morgan Kaufman 2008.
2. F. Y. Shih, Digital Watermarking and Stegonagraphy Fundamentals and Techniques, CRC press 2008.

3. Stefon Katzeubeisser, F. A. Petitolos, Information Hiding Techniques for Stegonagraphy and digital watermarking, Aatech House, London, 2008.

MCAE 307: QUANTUM COMPUTING [3-0-1]

Course Objectives: This course provides a foundation for quantum computing, post-quantum cryptography, and quantum machine learning. It covers the fundamental concepts of quantum mechanics, quantum algorithms, and their applications in various areas, including cryptography, cybersecurity, machine learning, finance, and the energy sector.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: describe the relevance of quantum mechanics to quantum computing.
CO2: describe and analyze quantum algorithms.
CO3: apply quantum optimization techniques in problem-solving in various areas, including cryptography and machine learning.

Syllabus:

Unit-I Introduction: Mathematical foundations: vectors, vector space, inner product; qubits, introduction to quantum mechanics and its relevance to quantum gates, superposition principle, and entanglement quantum parallelism and interference, no cloning theorem, quantum teleportation.

Unit-II Post-Quantum Security: Deutsch-Jozsa algorithm, Simon's algorithm, Bernstein-Vazirani, RSA algorithm and factorization attack on RSA, Shor's algorithm for integer factorization, Grover's algorithm for unstructured search, hash preimage attack with Grover's algorithm, Quantum Fourier transform and its applications, Harrow–Hassidim–Lloyd (HHL) algorithm, Quantum attack resistant Digital Signatures.

Unit-III Quantum Machine Learning and Optimization: Quantum machine learning (QML) models – QSVM, QNN, QCNN, Quantum Linear Regression, Variational Quantum Classifier (VQC), Quantum k-means clustering; kernel methods, Quantum Boltzmann Machines; Quantum optimization techniques: QAOA, quantum annealing.

Unit-IV: Introduction to Quantum Simulation Tools and Platforms: Google CIRQ, Amazon Braket, IBM Qiskit, Pennylane, Q#, Tensorflow quantum, Tket/pyket, XACC, Project Q, Quantum Development Kit (QDK).

Readings:

1. Elias F. Combarro, Samuel González-Castillo, and Alberto Di Meglio. **A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms,** Packt Publishing Ltd, 2023.
2. Noson S. Yanofsky and Mirco A. Mannucci, **Quantum Computing for Computer Scientists**. Cambridge University Press, 2008.
3. Douglas R. Stinson and Maura B. Paterson. **Cryptography, Theory and Practice,** CRC Press, 2019.
4. Santanu Pattanayak. **Quantum Machine Learning with Python: Using Cirq from Google Research and IBM Qiskit**. Apress, 2021.
5. Santanu Ganguly, **Quantum Machine Learning: An Applied Approach,** Apress, 2021.
6. https://docs.quantum.ibm.com/
7. https://quantumai.google/cirq/experiments/textbook_algorithms

MCAE 308: COMPUTER GRAPHICS [3-0-1]

Course Objectives: The course is designed to introduce students to the basic concepts and theory of computer graphics. The aim is to introduce students regarding various object rendering algorithms, projections, and latest concepts related to virtual reality.

Course Learning Outcomes:

On completing this course, the student will be able to:

CO1: acquire familiarity with the concepts and relevant mathematics of computer graphics.
CO2: ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping.
CO3: describe the importance of viewing and projections.
CO4: ability to design basic graphics application programs.
CO5: familiarize with fundamentals of animation and Virtual reality technologies.
CO6: be able to design applications that display graphic images to given specifications.

Syllabus:

UNIT I. Application Areas of Computer Graphics: Overview of graphics systems and devices. points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. filled area primitives, polygon filling algorithms. curve generation: Bezier and B-Spline Curves.

UNIT II 2-D Geometrical Transforms: Translation, scaling, rotation, reflection and shear transformations composite transforms, transformations between coordinate systems. 2-D Viewing: viewing pipeline, viewing coordinate reference frame, window to viewport coordinate transformation, viewing functions.

UNIT III Line Clipping Algorithms: Cohen-Sutherland and Cyrus Beck Line Clipping Algorithms, Sutherland–Hodgeman polygon clipping algorithm. 3-D object representation: polygon surfaces, quadric surfaces, spline representation.

UNIT IV 3-D Geometric Transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: viewing pipeline, viewing coordinates, view volume, general projection transforms and clipping.

UNIT V Visible Surface Detection Methods: Classification, back-face detection, depth-buffer, scanline, depth sorting, BSP-tree methods, area sub-division and octree methods illumination models and surface rendering methods: basic illumination models, polygon rendering methods computer animation: design of animation sequence, general computer animation functions key frame animation, animation sequence, motion control methods, morphing, mesh warping.

UNIT VI Virtual Reality: Basic concepts, classical components of VR system, types of VR systems, three-dimensional position trackers, navigation and manipulation interfaces, gesture interfaces. input devices, graphical rendering pipeline, haptic rendering pipeline, OpenGI rendering pipeline. applications of virtual reality.

Readings:

1. Donald Hearn, Pauline Baker, Warren Carithers, **Computer Graphics with Open GL**, Prentice Hall, 2010.
2. R. K Maurya, **Computer Graphics with Virtual Reality Systems**, Wiley, 2014

MCAE309: DIGITAL IMAGE PROCESSING [3-0-1]

Course Objectives: The course aims to cover core concepts in digital image processing. The course begins with the image enhancement techniques in the spatial and frequency domain, followed by the image morphological operations such as dilation, erosion, and hit-or-miss transformations. The course also covers image segmentation and image compression.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1 compare different techniques of image acquisition, enhancement, compression and segmentation.
CO2 choose appropriate feature extraction technique for an application..
CO3 compare and contrast merits of different image compression techniques
CO4 implement various image processing techniques.

Syllabus:

Fundamental Steps in Image Processing: Element of visual perception, a simple image model, sampling and quantization, some basic relationships between pixel, image geometry in 2D, image enhancement in the spatial domain.

Introduction to spatial and frequency methods: Basic gray level transformations, histogram equalization, local enhancement, image subtraction, image averaging, basic spatial, filtering, smoothing spatial filters, sharpening spatial filters.

Introduction to the Fourier transformation: Discrete fourier transformation, fast Fourier transformation, filtering in the frequency domain, correspondence between filtering in the spatial and frequency domain smoothing frequency-domain filters, sharpening frequency-domain filters, homomorphic filtering,

Some basic morphological algorithms: Line detection, edge detection, gradient operator, edge linking and boundary detection, thresholding, region-oriented segmentation, representation schemes like chain codes, polygonal approximations, boundary segments, skeleton of a region.

Introduction to Image Compression: JPEG, MPEG, Wavelets

Readings:

1. Rafael C. Gonzalez and Richard E.Woods, **Digital Image Processing**, Prentice–Hall of India, 2002
2. William K. Pratt, **Digital Image Processing: PIKS Inside** (3rd ed.), John Wiley & Sons, Inc., 2001
3. Bernd Jahne, **Digital Image Processing**, (5th revised and extended edition), Springer, 2002
4. S. Annadurai and R. Shanmugalakshmi, **Fundamentals of Digital Image Processing**, Pearson Education, 2007
5. M.A. Joshi, **Digital Image Processing: An Algorithmic Approach**, Prentice-Hall of India, 2006
6. B. Chanda and D.D. Majumder, **Digital Image Processing and Analysis**, Prentice-Hall of India, 2007

MCAE310 SOCIAL NETWORKS [3-0-1]

Course Objectives: The course aims to equip students with various social network analysis approaches to data collection, cleaning, and pre-processing of network data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:

CO1: identify different types of social networks and their characteristics.

CO2: implement and apply various social network analysis techniques, such as, influence maximization, community detection, link prediction, and information diffusion.

CO3: apply network models to understand phenomena such as social influence, diffusion of innovations, and community formation.

Syllabus:

Unit-I: Introduction to Social Network Analysis: Introduction to social network analysis, types of networks, nodes edges, node centrality, betweenness, closeness, eigenvector centrality, network centralization, assortativity, transitivity, reciprocity, similarity, degeneracy and network measure, networks structures, network visualization, tie strength, trust, understanding structure through user attributes and behavior.

Unit-II: Link Analysis and Link Prediction: Applications of link analysis, signed networks, strong and weak ties, link analysis and algorithms, page rank, personalized pagerank, divrank, simrank, pathsim. temporal changes in a network, evaluation link prediction algorithms, heuristic models, probabilistic models, applications of link prediction.

Unit-III: Community Detection: Applications of community detection, types of communities, community detection algorithms, disjoint community detection, overlapping community detection, local community detection, evaluation of community detection algorithms.

Unit-IV: Influence Maximization: Applications of influence maximization, diffusion models, independent cascade model, linear threshold model, triggering model, time-aware diffusion model, non-progressive diffusion model. influence maximization algorithms, simulation-based algorithms, proxy-based algorithms, sketch-based algorithms, community-based influence maximization, and context-aware influence maximization.

Unit-V: Multilayer Social Network: Multilayer social networks, formation of multilayer social networks, heuristic-based approaches, greedy approaches, centrality-based approaches, meta-heuristic approaches, path-based approaches, measuring multilayer social networks.

Readings:

1. Tanmoy Chakraborty, **Social Network Analysis,** Wiley India, 2021.
2. David Knoke and Song Yang, **Social Network Analysis,** SAGE publications, 2019.
3. Mark E. Dickison, Matteo Magnani and Luca Rossi, **Multilayer Social Networks,** Cambridge University Press, 2016.
4. Jennifer Golbeck, **Analyzing the Social Web**, Morgan Kaufmann, 2013.
5. Stanley Wasserman, and Katherine Faust. **Social Network Analysis: Methods and applications,** Cambridge University Press, 2012.
6. M.E.J. Newman, **Networks: An introduction,** Oxford University Press, 2010.
7. Wei Chen, Carlos Castillo and Laks V.S. Lakshmanan, **Information and Influence Propagation in Social Networks**, Springer, 2014
8. Virinchi Srinivas and Pabitra Mitra, **Link Prediction in Social Networks: Role of Power-law Distribution**, Springer International Publishing, 2016

MCAE311: CLOUD COMPUTING [3-0-1]

Course Objectives: This course aims to equip the students with parallel and distributed computing and cloud computing concepts. Students will learn about cloud computing's characteristics, benefits, and historical developments. They will learn cloud computing architecture, service models (IaaS, PaaS, SaaS), deployment models, and emerging paradigms like Edge Computing and Mobile Cloud Computing.

Course Learning Outcomes :

Upon successful completion of this course, a student will be able to:

CO1: describe cloud computing's characteristics, benefits, and historical developments, including distributed systems and virtualization.
CO2: compare and contrast cloud computing architectures, service models, and deployment models.
CO3: analyze cloud economics, address open challenges.
CO4: discuss emerging paradigms like edge computing and mobile cloud computing
CO5: develop a cloud comping application.

Syllabus:

Unit-I. Introduction: Introduction to parallel and distributed computing, cloud computing: characteristics and benefits; historical developments and evolution of cloud computing: distributed systems, virtualization, Web 2.0, service-oriented computing, utility computing.

Unit-II. Virtualization: Cloud computing reference model, characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud computing, pros and cons of virtualization, technology examples: Xen: paravirtualization, VMware: full virtualization, Microsoft Hyper-V.

Unit-III: Cloud Computing Architecture and Service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS); Deployment models: Public, Private, Hybrid, Community; IaaS: Introduction to IaaS, Resource Virtualization i.e. Server, Storage and Network virtualization; PaaS: Introduction to PaaS, Cloud platform & Management of Computation and Storage; SaaS: Introduction to SaaS, Cloud Services, Web services, Web 2.0, Web OS; Case studies related to IaaS, PaaS and SaaS, Economics of the cloud.

Unit-IV. Current Topics: Open Challenges in Cloud Computing; Introduction to emerging computing paradigms and research challenges: edge computing, mobile cloud computing, fog computing, etc.; Introduction to IoT cloud; study on simulators related to cloud computing and emerging computing paradigms.

Readings:

1. R. Buyya, C. Vecchiola, S. ThamaraiSelvi, **Mastering Cloud Computing**, McGraw Hill, 2013.

2. B. Sosinsky, **Cloud Computing Bible,** Wiley, 2010

3. K. Hwang, G. C. Fox, J. Dongarra, **Distributed and Cloud Computing: From Parallel Processing to the Internet of Things,** Morgan Kaufmann, 2011

MCAO 301: DATA MINING FOR BUSINESS APPLICATIONS (4+1)

Course Objectives: In this course, the objective is to explore the applications of data mining for business. The course should enable students to translate business problems into data hypotheses. Data wrangling and visualization followed by classification and clustering of data points are other intents of the course. The course also covers mining frequent itemsets from transactional data.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: Visualize data from a business perspective
CO2: Carry out data wrangling tasks, like handling missing data and data transformation
CO3: Map a business problem to data mining tasks like classification, clustering, and rule mining, and apply a suitable technique to it.
CO4: Use programming tools for the application of data mining tasks.

Syllabus:

Unit-I Handling Business Data: Translate business problems into data hypotheses, exploring and describing datasets, predictive and descriptive data mining techniques, use of visualizations to generate hypotheses.

Unit-II Data Wrangling: Prepare and clean data for analysis, identify solutions for managing missing data, data transformation, data reduction, discretization.

Unit-III Data Visualization: Generate insight with graphs, design visualizations to express data clearly.

Unit-IV Linear Regression: Identify relationships between variables, explain the linear model.

Unit-V Classification: Logistic regression, decision trees, Naïve Bayes, voting, confusion matrix, interpreting results, selecting a model to guide decisions.

Unit-VI Clustering: Clustering methods: centroid-based clustering, expectation maximization, hierarchical clustering, density-based methods, cluster validation methods, and metrics.

Unit-VII Association Rule Mining: Frequent item sets, closed and maximal item sets, Apriori algorithm for association rule mining.

Readings:

1. P. Tan, M. Steinbach and V. Kumar, **Introduction to Data Mining**, Addison Wesley, 2016.
2. J Zaki Mohammed and Wagner Meira, **Data Mining and Analysis: Fundamental Concepts and Algorithms,** Cambridge University Press, 2014.
3. Jiawei Han and Micheline Kamber, **Data Mining: Concepts and Techniques** (3rd ed.), Morgan Kaufmann, 2011.
4. Robert Layton, **Learning Data Mining with Python**, Second Edition.

MCAO 302: DATA SCIENCE USING PYTHON[3-0-1]

Course Objectives: The objective of this course is to analyze the data statistically and discover valuable insights from it. The course gives hands-on practice on predictive and descriptive modeling of the preprocessed data. In addition, the student also learns to apply mining association rules from the transactional data and mining text from the document will also be covered during the course.

Course Learning Outcomes:

On completion of this course, the student will be able to:

CO1: demonstrate proficiency with statistical analysis of data.
CO2: develop the ability to build and assess data-based models.
CO3: execute statistical analyses and interpret outcomes.
CO4: apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively.

Syllabus:

Unit-I Introduction: Introduction data acquisition, data preprocessing techniques including data cleaning, selection, integration, transformation, and reduction, data mining, interpretation.

Unit-II Statistical data modeling: Review of basic probability theory and distributions, correlation coefficient, linear regression, statistical inference, exploratory data analysis, and visualization.

Unit-III Predictive modeling: Introduction to predictive modeling, decision tree, nearest neighbor classifier, and naïve Bayes classifier, classification performance evaluation, and model selection.

Unit-IV Descriptive Modeling: Introduction to clustering, partitional, hierarchical, and density based clustering (k-means, agglomerative, and DBSCAN), outlier detection, clustering performance evaluation.

Unit-V Association Rule Mining: Introduction to frequent pattern mining and association rule mining, Apriori algorithm, measures for evaluating the association patterns.

Unit-VI Text Mining: Introduction of the vector space model for document representation, term frequency-inverse document frequency (tf-idf) approach for term weighting, proximity measures for document comparison, document clustering, and text classification.

Readings:

1. W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy and iPython, 2nd Ed., O'Reilly, 2017.
2. P. Tan, M. Steinbach, A Karpatne, and V. Kumar, Introduction to Data Mining, 2nd Edition, Pearson Education, 2018.
3. G. Grolemund, H. Wickham, R for Data Science, 1st Ed., O'Reilly, 2017.

MCAO 303: DATA ANALYSIS AND VISUALIZATION [3-0-1]

Course Objectives: The course develops student's competence in cleaning and analyzing data related to a chosen application. It also aims to develop skills in using various tools for data visualization and choosing the right tool for given data.

Course Learning Outcomes:

Upon successful completion of this course, a student will be able to:
CO1: use data analysis tools with ease.
CO2: load, clean, transform, merge, and reshape data.
CO3: create informative visualisations and summarise data sets.
CO4: analyse and manipulate time series data.
CO5: solve real world data analysis problems.

Syllabus

Unit 1 Introduction: Introduction to data science, exploratory data analysis and data science process.

motivation for using Python for data analysis, introduction to Python shell, iPython, and Jupyter Notebook; essential Python libraries: NumPy, pandas, matplotlib, SciPy, scikit-learn, statsmodels.

Unit 2 Introduction to Pandas: Arrays and vectorized computation, introductio to Pandas data structures, essential functionality, summarizing and computing descriptive statistics. data loading, storage and file formats. reading and writing data in text format, web scraping, binary data formats, interacting with web APIs, interacting with databases, data cleaning and preparation, handling missing data, data transformation, string manipulation.

Unit 3 Data Wrangling: Hierarchical indexing, combining and merging data sets reshaping and pivoting. data visualization Matplotlib: basics of Matplotlib, plotting with Pandas and Seaborn, other Python visualization tools

Unit 4 Data Aggregation and Group operations: Data grouping, data aggregation, general split-apply-combine, pivot tables and cross tabulation

Unit 5 Time Series Data Analysis: Date and time data types and tools, time series basics, frequencies and shifting, time zone handling, periods and periods arithmetic, resampling and frequency conversion, moving window functions.

Readings:

1. W. McKinney, **Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython**, 2nd edition, O'Reilly Media.
2. C. O'Neil and R. Schutt (2013). **Doing Data Science: Straight Talk from the Frontline,** O'Reilly Media.

<u>MCAE 304: STATISTICAL METHODS [3-0-1]</u>

<u>Course Objectives</u>: To equip students with the skills necessary to apply statistical methods for various applications.

<u>Course Learning Outcomes:</u>

On completing this course, the student will be able to:

CO1: apply descriptive statistical techniques to summarize and interpret data
CO2: apply inferential statistical methods, including hypothesis testing and confidence interval estimation.
CO3: perform and interpret simple and multiple linear regression analysis
CO4: apply principles of experimental design in the context of a problem

<u>Syllabus:</u>

Unit-1 Introduction: Descriptive statistics: measures of central tendency and variability, representation of data: stem and leaf diagram, histogram, boxplot, and ogive; bar diagram and its variations, Pie charts; probability distributions: discrete and continuous, joint and conditional probability; theory of attributes: coefficient of association and coefficient of colligation.

Unit-II: Statistical Inference: Parameter and statistic; sampling distributions, confidence intervals and margin of error, hypothesis testing; ANOVA, parametric tests: normal test, t-test, f-test; non-parametric tests: Chi-square test for goodness of fit, Mann-Whitney U test, Kruskal-Wallis test.

Unit-III Regression and Classification: Correlation: measure and significance, simple linear regression, multiple linear regression, one-way classification, analysis of variance, two-way classification, analysis of covariance, curvilinear regression, factorial experiments, Spearman's rank correlation coefficient.

Readings:

1. Robert S. Witte and John S. Witte, **Statistics**, John Wiley & Sons Inc; 11th edition, 2021
2. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, **An Introduction to Statistical Learning**, Springer, 2023.
3. G. W. Snedecor, W. G. Cochran, **Statistical Methods**, Iowa State University Press, 1973

4. John A. Rice, **Mathematical Statistics and Data Analysis**, Cengage, 2013