INDEX

KALINDI COLLEGE

SEMESTER – VII

Bachelor of Vocation- Web Designing

S. No.	Content	Page No.
1	**DISCIPLINE SPECIFIC CORE (DSC)** • **Algorithm and Design (DSC-19)**	**[4-6]**
2	**DISCIPLINE SPECIFIC ELECTIVE (DSE)** • **Machine Learning** • **Computer Networks** • **Theory of Computation**	**[6-8]** **[9-10]** **[11-12]**
3	**Dissertation on Major** **OR** **Dissertation on Minor** **OR** **Academic project/ Entrepreneurship**	

INDEX
KALINDI COLLEGE
SEMESTER – VIII
Bachelor of Vocation- Web Designing

Sl. No.	Content	Page No.
1	**DISCIPLINE SPECIFIC CORE (DSC)** **Advanced Java Programming(DSC-20)**	**[13-15]**
2	**DISCIPLINE SPECIFIC ELECTIVE (DSE)** **Compiler Design** **Cloud Computing** **Deep Learning**	 **[16-17]** **[18-19]** **[20-21]**
3	**Dissertation on Major** **OR** **Dissertation on Minor** **OR** **Academic project/ Entrepreneurship**	

ALGORITHMS AND DESIGN

Course title &Code	Credits	Credit distribution of thecourse			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Algorithms and Design	**4**	**3**	**0**	**1**	**Class XII Pass**	**NA**

COURSE OBJECTIVES

- Understand and apply linear data structures-List, Stack and Queue.
- Understand the graph algorithms.
- Learn different algorithms analysis techniques.
- Apply data structures and algorithms in real time applications Able to analyze the efficiency of algorithm.

SYLLABUS

UNIT I LINEAR DATA STRUCTURES (9 hours)

Introduction - Abstract Data Types (ADT) —Stack —Queue —Circular Queue - Double Ended Queue - Applications of stack —Evaluating Arithmetic Expressions - Other Applications - Applications of Queue - Linked Lists - Singly Linked List - Circularly Linked List - Doubly Linked lists —Applications of linked list -Polynomial Manipulation.

UNIT Il NON-LINEAR TREE STRUCTURES (9 hours)

Binary Tree —expression trees — Binary tree traversals —applications of trees —Huffman Algorithm - Binary search tree - Balanced Trees - AVL Tree - B-Tree - Splay Trees —Heap- Heap operations- - Binomial Heaps - Fibonacci Heaps- Hash set.

UNIT Ill GRAPHS (9 hours)

Representation of graph - Graph Traversals - Depth-first and breadth-first traversal ,Applications of graphs - Topological sort ,shortest-path algorithms – Dijkstra’s algorithm , Bellman-Ford algorithm ,Floyd's Algorithm, minimum spanning tree ,Prim's and Kruskal's algorithms.

UNIT IV ALGORITHM DESIGN AND ANALYSIS (9 hours)

Algorithm Analysis —Asymptotic Notations - Divide and Conquer —Merge Sort —Quick Sort - Binary Search - Greedy Algorithms —Knapsack Problem —Dynamic Programming —Optimal Binary Search Tree – Warshall's Algorithm for Finding Transitive Closure.

UNIT V ADVANCED ALGORITHM DESIGN AND ANALYSIS (9 hours)

Backtracking —N-Queen's Problem - Branch and Bound —Assignment Problem - P & NP problems — NPcomplete problems — Approximation algorithms for NP-hard problems — Traveling salesman problem Amortized Analysis.

REFERENCES:

1. Anany Levitin "Introduction to the Design and Analysis of Algorithms" Pearson Education, 1; 2015
2. E. Horowitz, S.Sahni and Dinesh Mehta, "Fundamentals of Data structures in C++",University Press, 2007
3. E. Horowitz, S. Sahni and S. Rajasekaran, "Computer Algorithms/C++",Second Edition, University Press, 2007
4. Gilles Brassard, "Fundamentals of Algorithms", Pearson Education 2015
5. Harsh Bhasin, "Algorithms Design and Analysis", Oxford University Press 2015
6. John R.Hubbard, "Data Structures with Java", Pearson Education, 2015
7. M. A. Weiss, "Data Structures and Algorithm Analysis in Java", Pearson Education Asia, 2013

DISCIPLINE SPECIFIC ELECTIVE – MACHINE LEARNING

Course title &Code	Credits	Credit distribution of thecourse			Eligibility criteria	Pre- requisite of the Course (if any)
		Lecture	Tutorial	Practical/ Practice		
Machine Learning	4	3	0	1	Class XII Pass	NA

Learning Objectives

The Learning Objectives of this course are as follows:

- This course will present foundations of Machine Learning algorithms, as well as their real-world applications.
- The course will cover two major learning approaches: supervised and unsupervised.

Learning outcomes

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

- Differentiate between supervised and unsupervised learning tasks.
- Normalize the data and perform outlier analysis.
- Execute various machine learning algorithms learnt in the course. Understand the concepts of regression, clustering and dimensionality reduction.

Syllabus

Unit I: Introduction to Machine Learning (8 hours)

Definition, History, Future and basic concepts of Machine Learning, Statistical learning vs machine learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Data measurement scales, Feature Engineering: normalizing data, missing value treatment, outliers. Creating graphs (bar/line/pie/boxplot/histogram, etc.), summarizing data, descriptive statistics, univariate analysis (distribution of data), bivariate analysis (cross tabs, distributions and relationships, graphical analysis), Splitting dataset into training and test set.

Unit II: Supervised Learning- I **(12 hours)**

Linear Regression: Regression with one variable, multiple variables, over-fitting, regularization. Regression evaluation metrics. Logistic Regression, Discriminant Analysis,

Unit III: Supervised Learning- II **(12 hours)**

k-nearest neighbor classifier, Naive Bayes classifier, Neural networks, Boosting, Support vector Machines, Decision trees, Bagging, Ensemble of Trees, Evaluating a Classification Model Performance, K-fold Cross-validation, ROC Curve

Unit IV: Unsupervised learning **(13 hours)**

clustering: Approaches for clustering, distance metrics, K-means clustering, hierarchical clustering, and feature selection methods. Dimensionality reduction (Principal Component Analysis). Association Rules Analysis.

Practical component (if any) [30 Hours]

Use Python for practical labs for Machine Learning.

List of Practical's:

1. Normalizing Data by Min-Max scaling
2. Split datasets into training and test sets and evaluate the decision models.
3. Summarizing Data through graphs and descriptive analysis.
4. Create Prediction Model for linear regression.
5. Make Prediction Model for logistic regression.
6. Make prediction models for Naïve Bayes Classifier Implement Decision Tree Implement SVM classification.
7. Perform K-means clustering.
8. Perform Principal Component analysis.

REFERENCES:

1. Alpaydin, E. (2020). Introduction to machine learning. MIT press.
2. Brownlee, J. (2018). Statistical methods for machine learning: Discover how to transform data into knowledge with Python. Machine Learning Mastery.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
4. Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.
5. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

COMPUTER NETWORKS

Course title & Code	Credits	Credit distribution of thecourse			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Computer Networks	4	3	0	1	Class XII Pass	NA

Course Objective:

The course provides a unified and fundamental view of the broad field of computer networks. Furthermore, the easy to understand and extremely relevant world of Computer Networking is introduced in a top down Approach. Introduction to intranets and intranet servers and browsers, networks and network-servers, LANs/WANs, Internet working technologies, the OSI reference model for networking protocols, CSMA/CD, TCP/IP implementation

Syllabus

Unit I: Introduction **(6 Hours)**

Introduction to computer networks, evolution of computer networks and its uses, Advantages and Disadvantages of Computer Network, reference models: OSI reference Models, TCP/IP Protocol Suit Networking fundamentals: Internet, Circuit switching vs Packet switching, ISPs, Delay and Loss in Packet Switched Networks

Unit II: Local Area Network **(4 Hours)**

LAN Architecture, LAN topologies- Bus/ Tree LAN, Ring LAN, Star LAN, Wireless LAN, Ethernet and Fast Ethernet, Token Ring

Unit III: Application layer and data link layer **(10 Hours)**

Application Layer Protocols: HTTP, FTP, SMTP, DNS

Data link layer design issues, Flow Control- Stop and Wait, Error Detection, Error Control, error detection and correction, data link layer protocols, sliding window protocols, example of data link protocol- HDLC .

Unit IV: Medium access layer **(5 Hours)**

Channel allocation problem, multiple access protocols, Introduction to ALOHA, CSMA/CD, CSMA/CA

Unit V: The network layer **(10 Hours)**

Introduction, Routers, Network layer concepts, shortest path routing, flooding, distance vector routing, link state routing (without algorithms), congestion control and quality of service, internetworking, IP, Ipv4 Addressing vs Ipv6

Unit VI: The transport layer **(10 Hours)**

The transport layer services, elements of transport protocols, TCP and UDP, Brief introduction to **presentation and session layer, E-mail**

References:

- Data Communication & networking: Forouzan, B. A.
- Data and Computer Communications, W. Stallings, Prentice Hall of India
- Computer Networks: Tanenbaum, Andrew S, Prentice Hall

THEORY OF COMPUTATION

Course title & Code	Credits	Credit distribution of thecourse			Eligibility criteria	Pre- requisite of the Course (if any)
		Lecture	Tutorial	Practical/ Practice		
Theory of Computation	**4**	**3**	**0**	**1**	**Class XII Pass**	**NA**

COURSE OBJECTIVES:

- To learn about fundamental concepts of finite automata and formal language
- To enhance student's ability to understand and solve mathematical proofs for computation and algorithm
- To learn about deterministic and non- deterministic machines.
- To design grammars and recognizers for different formal languages

LEARNING OUTCOMES:

- Students will have clear understanding of abstract models of computation.
- Students will be able to analyses and design the finite automata, pushdown automata, formal language and language.
- Students will be able to apply mathematical and formal techniques for solving problems in computer science.

Syllabus

UNIT –I Introduction to Finite Automata (9 Hours)

The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata. Applications of finite automata, Finite automata with Epsilon transitions.

UNIT – II Finite Automata and Regular Expressions (10 Hours)

Applications of Regular Expressions; Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Decision properties of regular languages; Equivalence and minimization of automata.

UNIT – III Context—free grammars (9 Hours)

Parse trees; Applications; Ambiguity in grammars and Languages. Definition of the Pushdown automata; the languages of a PDA; Equivalence of PDA's and CFG's.

UNIT –IV Deterministic Pushdown Automata (8 Hours)

Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFLs.

UNIT –V The Turing machine (8 Hours)

Programming techniques for Turing Machines, Extensions to the basics Turning machines, Turning machines and computers.

REFERENCES:

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman: Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Education, 2011.

2. John C Martin: Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw- Hill, 2007.

3. Daniel L.A. Cohen: Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, 2009.

4. Thomas A. Sudkamp: An Introduction to the Theory of Computer Science, Languages and Machines, 3rd Edition,Pearson Education, 2006.

ADVANCED JAVA PROGRAMMING

Course title &Cod	Credits	Credit distribution of thecourse			Eligibilitycriter	Pre- requisite ofthe course (if any)
		Lecture	Tutorial	Practical/ Practice		
Advanced Java Programming	**4**	**3**	**0**	**1**	**Class XII Pas**	**NA**

Course Objectives:

This course is designed to develop understanding of object oriented programming concepts like Classes, Objects, Inheritance Polymorphism using Java . The course provides understanding of multithreading and excepyion handling in Java. It also introduces how to create Java application using Graphical User Interface (GUI).

Course Outcomes:

On successful completion of the course, students will be able to
Understand the object-oriented concepts—Classes, Objects, Inheritance, Polymrphisn— fire proble Solving.
Create and handle multithreading.
Handle program exceptions
Handle input/output through files
Create Java application with Graphical User Interface (GUI)

SYLLABUS

Unit 1 (6 hrs)

Introduction To Swing, MVC Architecture, Applets, Applications and Pluggable Look and Feel, Basic swing components : Text Fields, Buttons, Toggle Buttons, Checkboxes, and Radio Buttons

Unit 2 (6 hrs)

Java database Programming, java.sql Package, JDBC driver, Network Programming With java.net Package, Client and Server Programs, Content And Protocol Handlers

Unit 3 (9 hrs)

RMI architecture, RMI registry, Writing distributed application with RMI, Naming services, Naming And Directory Services, Overview of JNDI, Object serialization and Internationalization.

Unit 4 (6 hrs)

J2EE architecture, Enterprise application concepts, n-tier application concepts, J2EE platform, HTTP protocol, web application, Web containers and Application servers

Unit 5 (9 hrs)

Server side programming with Java Servlet, HTTP and Servlet, Servlet API, life cycle, configuration and context, Request and Response objects, Session handling and event handling, Introduction to filters with writing simple filter
Application

Unit 6 (9 hrs)

JSP architecture, JSP page life cycle, JSP elements, Expression Language, Tag Extensions, Tag Extension API, Tag handlers, JSP Fragments, Tag Files, JSTL, Core Tag library, overview of XML Tag library, SQL Tag library and Functions Tag library

Reference Books:

1. Java 6 Programming, Black Book, Dreamtech
2. Java Server Programming, Java EE6 (J2EE 1.6), Black Book, Dreamtech
3. Advanced Java Technology, By M.T. Savaliya, Dreamtech

List of Practical

1. Write a Program in Java to implement Calculator using Swing technology
2. Write a Program that displays two textboxes for entering a students' Roll-no and Name with appropriate labels and buttons.
3. Write a Java program that makes a connection with database using JDBC and prints metadata of this connection
4. Include the database connectivity in the program no.1.2 to insert, update, delete and display of student information.
5. Write a java program for one way TCP communication for server and client, where server will response to client with current data and time.
6. Write a java program for two way TCP communication for server and client. It should look like a simple chat application
7. Write a java program for UDP Communication where client will send name of country and server will return the capital of that country.
8. Create a simple calculator application that demonstrates the use of RMI. You are not required to create GUI.
9. Create Servlet for login page, if the username and password is correct then prints message "Hello username" else a message"login failed".
10. Create Servlet that uses cookies to store the number of times a user has visited the servlet.
11. Create a Servlet for demo of KBC game. There will be continuous two or three pages with different MCQs. Each correct answer carries Rs. 10000. At the end as per user's selection of answers total prize he won should be declared. User should not be allowed to backtrack.

12. Create a Servlet that implements ServletContextAttributeListener interface such that a message dialog is displaued whenever an attribute is added or removed or replaced.
13. Create a Servlet filter that calculates server's response time and add it to response when giving it back to client.
14. Create a jsp that prints hello world.
15. Create jsp that prints current date and time.
16. Create a jsp that add and subtract two numbers.
17. Create a jsp for login module.
18. Create a web page that prints 1 to 10 using JSTL
19. Create a custom JSP tag that prints current date and time. Use this tag into JSP page.

COMPILER DESIGN

Course title &Code	Credits	Credit distribution of thecourse			criteria	Pre- requisite ofthe course (if any)
		Lecture	Tutorial	Practical/ Practice		
Compiler Design	4	3	0	1	Class XII Pass	NA

Course Objectives:

To teach concepts of language translation and phases of compiler design
To describe the common forms of parsers
To inculcate knowledge of parser by parsing LL parser and LR parser
To demonstrate intermediate code using technique of syntax directed translation
To Illustrate the various optimization techniques for designing various optimizing compilers.

Course Outcomes:

At the end of the course students will be able to:
Use compiler construction tools and describes the Functionality of each stage of compilation Process
Construct Grammars for Natural Languages and find the Syntactical Errors/Semantic errors during the compilations using parsing techniques
Analyze different representations of intermediate code.
Construct new compiler for new languages.
Participate in GATE, PGECET and other competitive examinations

SYLLABUS

UNIT -1 (9 hrs)

INTRODUCTION TO COMPILERS: Definition of compiler, interpreter and its differences, the phases of a compiler, role of lexical analyzer, regular expressions, finite automata, from regular expressions to finite automata, pass and phases of translation, bootstrapping, LEX-lexical analyzer generator. PARSING: Parsing, role of parser, context free grammar, derivations, parse trees, ambiguity, elimination of left recursion, left factoring, eliminating ambiguity from dangling-else grammar, classes of parsing, top down parsing - backtracking, recursive descent parsing, predictive parsers, LL(1) grammars.

UNIT – Il (9 hrs)

BOTTOM UP PARSING: Definition of bottom up parsing, handles, handle pruning, stack implementation of shift-reduce parsing, conflicts during shift-reduce parsing, LR grammars, LR parsers-simple LR, canonical LR(CLR) and Look Ahead LR (LALR) parsers, error recovery in parsing, parsing ambiguous grammars, YACC-automatic parser generator.

UNIT – Ill (9 hrs)

SYNTAX DIRECTED TRANSLATION: Syntax directed definition, construction of syntax trees, S- attributed and L-attributed definitions, translation schemes, emitting a translation.
INTERMEDIATE CODE GENERATION: intermediate forms of source programs—abstract syntax tree, polish notation and three address code, types of three address statements and its implementation, syntax directed translation into three-address code, translation of simple statements, Boolean expressions and flow-of-control statements.

UNIT-IV (9 hrs)

TYPE CHECKING: Definition of type checking, type expressions, type systems, static and dynamic checking of types, specification of a simple type checker, equivalence of type expressions, type conversions, overloading of functions and operators.
RUN TIME ENVIRONMENTS: Source language issues, Storage organization, storage-allocation strategies, access to non-local names, parameter passing, symbol tables and language facilities for dynamic storage allocation.

UNIT-V (9 hrs)

CODE OPTIMIZATION: Organization of code optimizer, basic blocks and flow graphs, optimization of basic blocks, the principal sources of optimization, the directed acyclic graph (DAG) representation of basic block, global data flow analysis.
CODE GENERATION: Machine dependent code generation, object code forms, the target machine, a simple code generator, register allocation and assignment, peephole optimization.

Reference Books:

1. Alfred V.Aho, Ravi Sethi, Jeffrey D.Uliman (2007), Compilers Principles, Techniques and Tools, 2nd edition, Pearson Education, New Delhi, India.
2. Alfred V. Aho, Jeffrey D. Uliman (2001), Principles of compiler design, Indian student edition, Pearson Education, New Delhi, India.
3. Kenneth C. Louden (1997), Compiler Construction— Principles and Practice, 1" edition, PWS Publishing.
4. K.L.P Mishra, N. Chandrashekaran (2003), Theory of computer science-Automata Languages and computation, 2™ edition, Prentice Hall of India, New Delhi, India.

CLOUD COMPUTING

Course title &Co	Credits	Credit distribution of thecourse			Eligibilitycriteria	Pre- requisite ofthe course (if any)
		Lecture	Tutorial	Practical/ Practice		
Cloud Computing	4	3	0	1	Class XII Pass	Operating System, Computer Networks

Course Objectives:

Learn the fundamental concepts of cloud computing, including its services, models, and deployments.

Course Outcomes:

Be able to apply cloud computing concepts in real-world scenarios, analyze cloud programming models, and manage resources
Understand the theoretical background for cloud computing, develop applications for cloud environments, and use IaaS and PaaS software to realize cloud infrastructures
The evolution of cloud computing and its applicability

SYLLABUS

UNIT I (3 weeks)

Cloud Computing fundamentals: Cloud Delivery models, The SPI Framework, Cloud Software as a Service (SaaS) , Cloud Platform as a Service(PaaS), Cloud Infrastructure as a Service(laaS), Cloud deployment models, Public Clouds, Community Clouds, Hybrid Clouds, Alternative Deployment models, Expected benefits.

UNIT – II (3 weeks)
Cloud Computing Architecture: Cloud Delivery models, The SPI Framework, Cloud Software as a Service (SaaS) , Cloud Platform as a Service(PaaS), Cloud Infrastructure as a Service(laaS), Cloud deployment models, Public Clouds, Community Clouds, Hybrid Clouds, Alternative Deployment models, Expected benefits.

UNIT-III (3 weeks)
Cloud Computing Software Security fundamentals: Cloud Information Security Objectives, Confidentiality, Integrity, Availability, Cloud Security Services, Relevant Cloud Security Design Principles, Secure Cloud Software Requirements, Secure Development practices,

Approaches to Cloud Software Requirement Engineering, Cloud Security Policy Implementation.

UNIT – IV (3 weeks)

Cloud Computing Risk Issues: The CIA Traid, Privacy and Compliance Risks, Threats to Infrastructure, Data and Access Control, Cloud Access Control Issues, Cloud Service Provider
Risks.

Cloud Computing Security challenges: Security Policy Implementation, Policy Types, and Computer Security Incident Response Team (CSIRT).

UNIT – V (3 weeks)

Cloud Computing Security Architecture: Architectural Considerations, General Issues, Trusted Cloud Computing, Secure Execution environments and Communications, Micro architectures, Identity Management and Access Control, Autonomic Security.

Reference Books:

1. Ronald L. Krutz, Russell Dean Vines, "Cloud Security Acomprehensive Guide tosecure Cloud Computing" Wiley.
2. John W. itinghouse james F.Ransome, "Cloud Computing Implementation, Management and Security" ,CRC Press.
3. Borko Furht. Armando Escalante, "Handbook of Cloud Computing", Springer
4. Charles Badcock, "Cloud Revolution" ,TMH

DEEP LEARNING

Course title & Code	Credits	Credit distribution of thecourse			Eligibilitycriteria	Pre- requisite ofthe course (if any)
		Lecture	Tutorial	Practical/ Practice		
Deep Learning	4	3	0	1	Class XII Pass	NA

Course Objectives:

Learn the fundamental concepts of Deep learning, real world Problem.

Course Outcomes:

On completion of this course, the student twill be able to:
Describe the feed forward and deep networks
Design single and multi-layer feed-forward deep networks and tune various hyper- parameters.
Analyze performance of deep networks.

Syllabus

Unit I (9 hrs)

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

Unit II (8 hrs)

Neural networks: Feedforward neural networks, deep networks, regularizing a deep network, model exploration, and hyper parameter tuning.

Unit III (10 hrs)

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

Unit IV (10 hrs)

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

Unit V (8 hrs)

Autoencoders: Under complete autoencoders, regularized-autoencoders, sparse autoencoders, denoising

autoencoders ,representational power, layer, size, and depth of autoencoders, stochastic encoders and decoders.

Reference Books:

1. Ian Good fellow, **Deep Learning**, MITPress,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, Dong Yu ,**Deep Learning: Methods and Applications**, Now Publishers Inc,2009