Sister Nivedita University

DG 1/2 New Town, Kolkata – 700156
www.snuniv.ac.in

OBE-RELATED ACADEMIC CURRICULUM

School of Science & Technology

Department of Computer Science

Syllabus

for

Master of Technology (M. Tech)

Computer Science and Engineering

Regulations (R24)

R24–25 Academic Session

VISION

To be a centre of excellence in computing, AI, and data-driven technologies, recognized for impactful research and innovation, while producing globally competent, ethical professionals and entrepreneurs who lead digital transformation across industries, and advancing interdisciplinary solutions that serve societal needs and support sustainable national and regional development.

MISSION

1. Deliver an outcome-based curriculum that blends strong computing fundamentals with emerging technologies through labs, projects, and real-world problem solving.

2. Foster a culture of research, innovation, and industry collaboration, entrepreneurship support, and modern infrastructure.

3. Nurture professionalism, ethics, inclusivity, effective communication, leadership, and lifelong learning to prepare graduates for responsible societal impact.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO1:

Advanced Technical and Research Excellence: Graduates of the M.Tech program in Computer Science and Engineering will demonstrate advanced proficiency in core and emerging areas of computing such as algorithms, distributed systems, data engineering, artificial intelligence, and software systems. They will be capable of formulating, analyzing, and solving complex engineering problems, conducting independent research, and applying rigorous scientific methods to develop innovative, high-quality technical solutions.

PEO2:

Career Advancement and Leadership: Graduates will achieve successful careers in academia, research organizations, industry, and entrepreneurial ventures by leveraging their deep technical knowledge and analytical skills. They will be prepared to take on leadership roles, manage multidisciplinary teams, drive technology-driven initiatives, and contribute strategically to the growth and competitiveness of their organizations.

PEO3:

Innovation, Entrepreneurship and Interdisciplinary Collaboration: Graduates will demonstrate the capability to innovate and translate research outcomes into practical applications, products, or services that address real-world challenges. They will effectively collaborate across disciplines, engage with stakeholders from diverse domains, and pursue entrepreneurial opportunities that foster technological advancement and societal benefit.

PEO4:

Ethical, Social and Environmental Responsibility: Graduates will uphold high standards of ethics, integrity, and professional conduct in their research and practice of computer science and engineering. They will recognize the societal, legal, security, and environmental implications of computing technologies and contribute responsibly towards sustainable development, inclusivity, and the well-being of the community at local, national, and global levels.

PROGRAM SPECIFIC OBJECTIVES (PSO)

PSO1:

Advanced Computing and System Design Proficiency: Graduates will be able to architect, model, analyze, and optimize complex computer systems and applications, including high-performance computing, cloud/edge infrastructures, and secure, scalable software systems, using advanced tools, frameworks, and engineering practices.

PSO2:

Intelligent Systems, Data Analytics and Emerging Technologies: Graduates will demonstrate the ability to design and implement intelligent systems by applying machine learning, deep learning, data analytics, and related techniques to large-scale, heterogeneous data. They will effectively employ emerging technologies in areas such as IoT, cyber-physical systems, and blockchain to build robust and innovative solutions.

PSO3:

Research, Innovation and Scholarly Contribution in CSE: Graduates will be able to identify research gaps, formulate research problems, develop and validate novel models, algorithms, or frameworks, and disseminate their findings through high-quality publications, patents, prototypes, or open-source contributions that advance the state of the art in computer science and engineering.

PROGRAM OUTCOMES (PO)

PO1: An ability to independently carry out research /investigation and development work to solve practical problems.

PO2: An ability to write and present a substantial technical report/document.

PO3: Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

Credit Definition

Type	Duration (in Hour)	Credit
Lecture (L)	1	1
Tutorial (T)	1	1
Practical (P)	2	1
Sessional (S)	2	1

Total Credit Distribution

Semester	Credits						Credits/Semester
	CC/PCC	DSE/PEC	USC/MUS	SEC/PSE	AECC/HSM	PROJECT	
1	16	8	2	2	0	0	28
2	4	13	2	0	0	3	22
3	8	0	0	0	2	8	18
4	0	0	0	2	0	16	18
Credits/Course	**28**	**21**	**4**	**4**	**2**	**27**	**86**

Category Definition

Definition of Category/Type	Abbreviation
Core Course/ Professional Core Course	**CC/PCC**
Discipline Specific Elective/ Professional Elective Course	**DSE/PEC**
Mandatory/ University Specified (Environmental Sc./ Induction Training/ Indian Constitution/ Foreign language)	**USC/MUS**
Skill Enhancement Course/ Program Specific Elective	**SEC/PSE**
Ability Enhancement Compulsory Courses/ Humanities & Social Science including Management	**AECC/HSM**
Project	**PROJECT**

FIRST YEAR

SEMESTER-I

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Mathematical Foundations of Computer Science		DSE/PEC	4	4	0	0	0
2	Advanced Operating System		CC/PCC	4	3	1	0	0
3	Advanced Data Structures		CC/PCC	4	3	1	0	0
4	Object Oriented Analysis and Design		CC/PCC	4	3	1	0	0
5	Distributed Systems/ Data Preparation and Analysis		DSE/PEC	4	3	1	0	0
6	Foreign Language I		USC/MUS	2	2	0	0	0
7	Research Methodology and IPR		SEC/PSE	2	2	0	0	0
Practical								
8	Object Oriented Analysis and Design Lab		CC/PCC	2	0	0	4	0
9	Advanced Data Structures Lab		CC/PCC	2	0	0	4	0
			Total Credit:	28				

SEMESTER-II

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	High Performance Computing		CC/PCC	4	3	1	0	0
2	Advanced Software Engineering/ Web Analytics and Development		DSE/PEC	4	3	1	0	0
3	Fog and Edge Computing/ Soft Computing		DSE/PEC	4	3	1	0	0
4	Machine Learning/ Quantum Computing		DSE/PEC	4	3	1	0	0
5	Foreign Language II		USC/MUS	2	2	0	0	0
6	Mini Project with Seminar / Industry Research Project		PROJECT	3	0	0	0	6
Practical								
7	Machine Learning Lab/ Quantum Computing Lab		DSE/PEC	1	0	0	2	0
			Total Credit:	22				

SECOND YEAR

SEMESTER-III

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Elective I		CC/PCC	4	3	1	0	0
2	Elective II		CC/PCC	4	3	1	0	0
3	Technical Scientific Writing		AECC/HSM	2	0	0	0	4
4	Customer Relationship Management using Salesforce		AECC/HSM	0	1	0	2	0
5	Research Project I		PROJECT	8	0	0	0	16
			Total Credit:	18				

Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1		Bioinformatics
2		Information Theory Coding
3		Mobile Computing
Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1		Human Computer Interaction
2		Big Data Analytics
3		Industry 5.0

SEMESTER-IV

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Dissertation/ Research Project II		PROJECT	16	0	0	0	32
2	Grand Viva		SEC/PSE	2	2	0	0	0
			Total Credit:	18				

FIRST YEAR

SEMESTER-I

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Mathematical Foundations of Computer Science		DSE/PEC	4	4	0	0	0
2	Advanced Operating System		CC/PCC	4	3	1	0	0
3	Advanced Data Structures		CC/PCC	4	3	1	0	0
4	Object Oriented Analysis and Design		CC/PCC	4	3	1	0	0
5	Distributed Systems/ Data Preparation and Analysis		DSE/PEC	4	3	1	0	0
6	Foreign Language I		USC/MUS	2	2	0	0	0
7	Research Methodology and IPR		SEC/PSE	2	2	0	0	0
Practical								
8	Object Oriented Analysis and Design Lab		CC/PCC	2	0	0	4	0
9	Advanced Data Structures Lab		CC/PCC	2	0	0	4	0
			Total Credit:	28				

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Mathematical Foundations of Computer Science	**Course Credit:** 04[4-0-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXXXXX	**Semester:** 1^{st}

➠ Learning Objectives:

On completion of the course, student will be able to: apply the knowledge of counting techniques and graph theory, linear algebra, abstract algebra to solve complex engineering problem.

➠ Prerequisite:

Before learning the concepts of Mathematical Foundations for Computer Science, you should have a basic knowledge of set, relation, mapping, matrix etc.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Functional Logic	6	13%
Module-II: Number Theory	6	13%
Module-III: Abstract Algebra	8	17%
Module-IV: Linear Algebra	10	21%
Module-V: Probability Theory	8	17%
Module-VI: Graphs Theory	10	21%

➠ Syllabus Outline: .....

Module I: Functional Logic: [6L]

Proposition Logic, Resolution Proof system, Predicate logic. Proposition, propositional variables, combination of propositions, Conjunction, Disjunction, Negation and their truth table, derived connectors. Conditional Connectives, Implication,Converse, Contrapositive,Inverse,Bi conditional statements with truth table,Logical Equivalence, Tautology, Normalforms-CNF, DNF; Predicates and Logical Quantifications

of propositions and related examples.

Module II: Number Theory: [6L]

Division Algorithm, Euclid's Algorithm, Prime numbers, Coprime, Congruences, Fermat's theorem, Euler function, Chinese remainder theorem.

Module III: Abstract Algebra: [8L]

Binary Operation, Semigroup, Monoid, Groups, subgroups, cyclic groups, permutation group, cosets and normal subgroups, Lagrange's theorem, Group homomorphism, Isomorphism theorems.

Module IV: Linear Algebra: [10L]

Matrix, Determinant, Rank, System of equations, Cramer's Rule, Gauss Elimination, Matrix Inversion Method, Vector Space, Linear dependence, Linear independence, Linear Span, Basis, Dimension, Subspaces, Linear Transformations, Matrix Representation, Co-ordinate Vector, Rank-Nullity Theorem, Linear Operator, Eigen values, Eigenvectors.

Module V: Probability Theory: [8L]

Counting, Probability, Discrete random variable, Continuous random variable, Moment generating function, Markov's inequality, Chebyshev's inequality, The geometric and binomial distributions, The tail of the binomial distribution, Poisson distribution, exponential distribution and normal distribution.

Module VI: Graphs Theory: [10L]

Basics of graph theory, graph isomorphism, complete graph, bipartite graph, adjacency matrix, incidence matrix, walk, path, trail, Euler tours, planar graphs, Hamiltonian graphs, Euler's formula, applications of Kuratowski's theorem, graph colouring, chromatic polynomials, trees, weighted trees, spanning tree, Minimum spanning tree, properties of trees, Algorithms: Dijkstra's Algorithm for shortest path problem, Determination of minimal spanning tree using Kruskal's and Prim's algorithms, the max-flow min-cut theorem.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. Topics in Algebra, I. N. Herstein, John Wiley and Sons.
2. Digital Logic & Computer Design, M. Morris Mano, Pearson.

3. Linear Algebra, Stephene H. Friedberg, Arnold J. Insel and Lawrence E. Spence.
4. Elements of Discrete Mathematics, (Second Edition) C. L. LiuMcGraw Hill, New Delhi.
5. Graph Theory with Applications, J. A. Bondy and U. S. R. Murty, Macmillan Press, London.
6. Mathematical Logic for Computer Science, L. Zhongwan, World Scientific, Singapore.

Reference Books:

1. Introduction to linear algebra. Gilbert Strang.
2. Introductory Combinatorics, R. A. Brualdi, North-Holland, New York.
3. Graph Theory with Applications to Engineering and Computer Science, N. Deo, Prentice Hall, Englewood Cliffs.

Course Outcome (CO):

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

CO1: *Understand* the fundamentals of Propositional Logic

CO2: *Identify* truth tables and logical operators to analyse problems.

CO3: *Understand* the fundamental theorems of Group theory.

CO4: *Understand* the fundamental concepts in graph theory.

CO5: *Apply* the knowledge of Boolean algebra in switching circuits and use Max-flow Min-cut theorem, Ford and Fulkerson algorithm to design complex engineering problems.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	-	1	-
CO2	3	2	-	-	1	-
CO3	2	1	-	1	-	-
CO4	2	1	1	2	1	-
CO5	3	2	1	3	1	1

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ADVANCED OPERATING SYSTEM

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Advanced Operating System	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 1^{st}

➠ Learning Objectives:

After completing this course, students will be able to:

1. Learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open-source operating systems).
2. Hardware and software features that support these systems.

➠ Prerequisite:

Students should have a solid understanding of basic operating system concepts, including processes, threads, CPU scheduling, memory management, file systems, and synchronization; prior experience with system programming, foundational knowledge of computer architecture, including interrupts, I/O systems; basic understanding of distributed systems, networking fundamentals, and concurrent programming; familiarity with data structures and algorithms for implementing OS components.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	16%
Module-II: Processes and processors in distributed systems	8	17%
Module-III: Distributed File Systems	8	17%
Module-IV: Distributed Shared Memory	8	17%
Module-V: Distributed Web-based Systems	6	12%
Module-VI: Distributed Deadlock Detection	10	21%

Syllabus Outline:

Module I: Introduction: [8L]

Overview, Functions of an Operating System, Design Approaches, Types of Advanced Operating System - Synchronization Mechanisms, Concept of a Process, Concurrent Processes, The Critical Section Problem, Other Synchronization Problems, Axiomatic Verification of Parallel Programs - Process Deadlocks - Models of Deadlocks, Resources, System State, Necessary and Sufficient conditions for a Deadlock.

Module II: Processes and processors in distributed systems: [8L]

Threads, system model, processor allocation, scheduling in distributed systems: Load balancing and sharing approach, fault tolerance, real time distributed systems, Process migration and related issues.

Module III: Distributed File Systems: [8L]

Introduction, features & goal of distributed file system, file models, file accessing models, file sharing semantics, file caching scheme, and file replication, fault tolerance, trends in distributed file system, case study.

Module IV: Distributed Shared Memory: [8L]

Introduction, general architecture of DSM systems, design and implementation issues of DSM, granularity, structure of shared memory space, consistency models, replacement strategy, thrashing

Module V: Distributed Web-based Systems: [6L]

Replication: Web Proxy Caching, Replication for Web Hosting Systems, Replication of Web Applications Architecture, Processes, Communication, Naming, Synchronization, Consistency and

Module VI: Distributed Deadlock Detection: [10L]

Preliminaries, Deadlock Handling Strategies in Distributed Systems, Issues in Deadlock Detection and Resolution, Control Organizations for Distributed Deadlock Detection, Centralized- Deadlock – Detection Algorithms, Distributed Deadlock Detection Algorithms, Hierarchical Deadlock Detection Algorithms

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Distributed Operating Systems Concepts and Design, Pradeep K. Sinha, PHI

2. Distributed Systems: Concepts and Design by George Coulouris, Jean Dollimore,Tim Kindberg, Pearsonn
3. Distributed Operating Systems by Andrew S Tannebaum, Pearson.

Reference Books:

1. Distributed Computing by Sunita Mahajan & Seema Shah OXFORD
2. Distributed Systems: Principles and Paradigms by Andrew S Tanebaum, Maarten Van Steen, PHI
3. Distributed Computing, Fundamentals, Simulations and Advanced topics, 2nd Edition, Hagit Attiya and Jennifer Welch,Wiley India

Course Outcome (CO):

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

CO1: *Explain* the functions, design approaches and types of advanced/distributed operating systems, and summarize process synchronization and deadlock conditions in distributed environments.

CO2: *Analyze* process and thread management in distributed systems, including processor allocation, load balancing, scheduling, fault tolerance and process migration strategies.

CO3: *Examine* and *compare* distributed file system and distributed shared memory architectures in terms of models, consistency, caching, replication, granularity and performance.

CO4: *Design* suitable distributed web-based system architectures by applying concepts of processes, communication, naming, synchronization, replication and consistency.

CO5: *Evaluate* and *differentiate* deadlock handling and distributed deadlock detection algorithms with respect to their control organization, overhead and correctness.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	3	3	-	1
CO2	3	1	3	3	-	2
CO3	3	2	3	3	2	3
CO4	3	2	3	3	3	2
CO5	3	2	3	2	-	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ADVANCED DATA STRUCTURES

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Advanced Data Structures	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 1^{st}

➠ Learning Objectives:

After completing this course, students will be able to:

1. Impart the basic concepts of data structures and algorithms.
2. Understand concepts about searching and sorting techniques.
3. Understand basic concepts about stacks, queues, lists, trees and graphs.
4. Understanding about writing algorithms and step by step approach in
5. Solve problems with the help of fundamental data structure

➠ Prerequisite:

Students should have a solid understanding of basic data structures (arrays, linked lists, stacks, queues, trees, graphs), algorithmic problem-solving skills, and foundational knowledge of time–space complexity analysis, typically gained from an introductory course in Data Structures and Algorithms.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to Data Structures	6	12.5%
Module-II: Structures and Arrays	6	12.5%
Module-III: Linear Data Structure	12	25%
Module-IV: Non-Linear Data Structure	12	25%
Module-V: Sorting Algorithms	6	12.5%
Module-VI: Searching and Hashing	6	12.5%

➠ Course Content/ Syllabus Table:

Module I: Introduction to Data Structures [6L]

Introduction, Definition, Classification of Data Structure, Description of Various Data Structures, Memory Allocations in C, Algorithms, Algorithm Performance, Algorithm Analysis, Categories of Algorithms, Data Structure operations, Abstract Data Types.

Module II: Structures and Arrays: [6L]

Why Use Structures, Declaring a Structure, Accessing Structure Elements, How Structure Elements are Stored, Array of Structures, Additional Features of Structures, Uses of Structures. Introduction, One Dimensional Array, Initializing One Dimensional Arrays, Accessing One Dimensional Arrays Elements, Implementation of One Dimensional Array in Memory, Passing Array to Functions, Insertion in One Dimensional Array, Deletion of Element One Dimensional Arrays, Traversing of an Array, Multi-Dimensional Arrays, Initializing a Two Dimensional Array, Accessing Two Dimensional Arrays Elements, Implementation of Two Dimensional Array in Memory, Pointers and Arrays, Array of Pointers, Array of Structures, Array within the Structure, Limitation of Linear Array.

Module III: Linear Data Structure: [12L]

Introduction, Stack Implementation, Operation on Stack, Stack Terminology, Algorithms for Push and Pop, Implementing Stack Using Pointers, Application of Stacks, Algorithm for Converting Infix to Expression to Postfix Form, Converting Infix to Expression to Prefix Form,Algorithm to Evaluate to Postfix Expression, Binary Expression Tree.Introduction, Implementation of Queue, Operation on a Queue, Algorithm for insertion and deletion on Queue (Using Array), Limitation of Simple Queue, Algorithm for insertion and deletion on Queue (Using Pointers), A Circular Queue, Double Ended Queues(deque), Priority Queue, Application of Queues, Multiple Queues. Introduction, Linked Lists, Key terms, Representation of linear linked list, Operations on linked list, Types of linkedlist, Singlylinkedlist, Circular linkedlist, Doubly linkedlist, Circular doubly linkedlist, Application: Addition of two polynomials, Generalized Linkedlist.

Module IV: Non-Linear Data Structure: [12L]

Introduction, Tree, Binary trees, Binary trees representation, Creation of Binary tree, Operation on Binary trees, Technique of converting an expression into binary tree, Binary search tree, Threaded Binary Trees, Height balanced binary tree, B-Tree, B+Tree, Extended Binary tree. Introduction, Defining graph, Basic terminology, Graph representation, Graph Implementation, Shortest path problem, Minimum Spanning tree, Shortest path algorithm.

Module V: Sorting Algorithms: [6L]

Introduction, Sorting, Bubble sort, Selection sort, Insertion sort, Quick sort

Module VI: Searching and Hashing: [6L]

Introduction, Linear searching, Binary searching, Hashing, Terms associated with hash tables, Bucket overflow, Advantages of chaining

➠ Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books:

Text Books:

1. Fundamentals of Data Structures in C, E.Horowitz- S.Sahni, Galgotia-2006
2. Data Structures and Algorithm Analysis in C, M.A.Weiss, Pearson Education-Fourth Edition

Reference Books:

1. Data Structures, Algorithms and Applications in C, Sartaj Sahni, University Press
2. Data Structures using C by Yedidyah Langsam, Moshe J. Augenstein and Aron M. Tananbaum, PHI.2002
3. Karumanchi, N. (2016). Data Structures and Algorithms Made Easy. CareerMonk Publications.
4. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). Data Structures and Algorithms in Java (6th ed.). Wiley.

➠ Course Outcome (CO):

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

CO1: *Explain* the fundamental concepts, classifications, memory allocation techniques, and abstract data types used in data structures.

CO2: *Apply* structures, arrays, and pointer-based representations to implement basic data structure operations in C.

CO3: *Implement* linear data structures such as stacks, queues, and linked lists, including expression conversion/evaluation and polynomial operations.

CO4: *Aanalyze* and *construct* non-linear data structures (trees, BST, AVL, B/B+ trees, graphs) and apply fundamental graph and tree algorithms.

CO5: *Evaluate* and *implement* efficient searching, sorting, and hashing techniques to solve computational problems.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	1	3	2
CO2	3	3	2	1	3	3
CO3	3	3	3	1	3	3
CO4	3	3	3	1	3	3
CO5	3	3	3	1	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

OBJECT ORIENTED ANALYSIS AND DESIGN

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Object Oriented Analysis and Design	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 1^{st}

Learning Objectives:

After completing this course, students will be able to:

1. Develop proficiency in object-oriented concepts, UML modeling techniques, and design patterns to analyze and design complex software systems.
2. Apply OOAD principles and methodologies to real-world case studies, emphasizing the entire software development lifecycle from requirements gathering to implementation.
3. Gain practical skills in translating object-oriented designs into code, implementing design patterns, and conducting various levels of testing for object-oriented systems.

Prerequisite:

Students should have prior knowledge of object-oriented programming (OOP) concepts along with basic software engineering principles and familiarity with at least one OOP language like Java, C++, or Python. A foundational understanding of algorithms, data structures, and general software development processes is also expected.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to OOAD	6	12%
Module-II: Design Patterns	10	21%
Module-III: Case Study	10	21%
Module-IV: Applying Design Patterns	6	12%
Module-V: Coding And Testing	8	17%
Module-VI: Component Diagram Model and Deployment Model	8	17%

Syllabus Outline:

Module I: Introduction to OOAD : [6L]

Unified Process – UML diagrams – Use Case – Class Diagrams– Interaction Diagrams – State Diagrams – Activity Diagrams – Package, component and Deployment Diagrams.

Module II: Design Patterns: [10L]

GRASP: Designing objects with responsibilities – Creator – Information expert – Low Coupling – High Cohesion – Controller – Design Patterns – creational – factory method – structural – Bridge – Adapter – behavioral – Strategy – observer.

Module III: Case Study : [10L]

Case study – the Next Gen POS system, Inception -Use case Modeling – Relating Use cases – include, extend and generalization – Elaboration – Domain Models – Finding conceptual classes and description classes – Associations – Attributes – Domain model refinement – Finding conceptual class Hierarchies – Aggregation and Composition.

Module IV: Applying Design Patterns: [6L]

System sequence diagrams – Relationship between sequence diagrams and use cases Logical architecture and UML package diagram – Logical architecture refinement – UML class diagrams – UML interaction diagrams – Applying GoF design patterns.

Module V: Coding And Testing: [8L]

Mapping design to code – Testing: Issues in OO Testing – Class Testing – OO Integration Testing – GUI Testing – OO System Testing.

Module VI: Component Diagram Model and Deployment Model: [8L]

Physical Aspect, Logical Aspect, Connections and Dependencies, User face, Initial DB design in a UML environment, Processors, Connections, Components, Tasks, Threads, Signals and Events.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Craig Larman, "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2005.
2. Simon Bennett, Steve Mc Robb and Ray Farmer, "Object Oriented Systems Analysis and Design Using UML", Fourth Edition, Mc-Graw Hill Education, 2010.
3. Erich Gamma, and Richard Helm, Ralph Johnson, John Vlissides, "Design patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.

Reference Books:

1. Martin Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third edition, Addison Wesley, 2003.
2. Paul C. Jorgensen, "Software Testing: - A Craftsman's Approach", Third Edition, Auerbach Publications, Taylor and Francis Group, 2008.

Course Outcome (CO):

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

CO1: *Define* fundamental concepts of Object Oriented Analysis and Design including Unified Process, UML diagrams, Use Case, Class, Interaction, State, Activity, Package, Component and Deployment diagrams..

CO2: *Explain* GRASP principles and classify basic GoF design patterns such as Factory Method, Adapter, Bridge, Strategy and Observer with suitable illustrations..

CO3: *Implement* use case modeling and domain modeling techniques for the Next Gen POS system by constructing conceptual classes, associations, attributes and relationships such as include, extend and generalization..

CO4: *Analyze* the relationship between use cases, sequence diagrams, logical architecture and UML interaction diagrams for refining system design using GoF design patterns..

CO5: *Design* and *evaluate* object-oriented systems by integrating testing strategies, component modeling and deployment design involving processors, components, threads, events and database structures in UML environment.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	1	1	2	2	1	1
CO2	2	1	2	2	2	1
CO3	3	2	3	3	2	2
CO4	3	2	3	3	2	2
CO5	3	2	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

DISTRIBUTED SYSTEMS

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Distributed Systems	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 1^{st}

Learning Objectives:

After completing this course, students will be able to:

1. Understand fundamental concepts, characteristics, and architectural models of distributed systems.
2. Learn communication models, logical clocks, message ordering, and synchronization challenges in distributed execution.
3. Learn distributed algorithms for mutual exclusion, snapshot recording, deadlock detection, recovery, and consensus.
4. Learn distributed mutual exclusion and deadlock detection algorithms.
5. Understand the significance of agreement, distributed system performance, fault tolerance techniques and recovery protocols in Distributed Systems.
6. Learn the characteristics of peer-to-peer and distributed shared memory systems

Prerequisite:

Students should have prior knowledge of operating systems concepts (processes, threads, synchronization, deadlocks), computer networks (network layers, communication protocols, message passing), and basic algorithms and data structures. Familiarity with programming in C, Java, or Python, along with foundational understanding of concurrency and system architecture, is also required.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	16.66%
Module-II: Design issues and challenges	8	16.66%
Module-III: Message Ordering & Snapshots	8	16.67%
Module-IV: Distributed Mutex & Deadlock	8	16.67%
Module-V: Recovery & Consensus	8	16.67%
Module-VI: P2P & Distributed Shared Memory	8	16.67%

Syllabus Outline:

Module I: Introduction: [8L]

Introduction: Definition – Relation to computer system components – Motivation –Relation to parallel systems – Message-passing systems versus shared memory systems – Primitives for distributed communication – Synchronous versus asynchronous executions.

Module II: Design issues and challenges: [8L]

A model of distributed computations: A distributed program –A model of distributed executions –Models of communication networks –Global state – Cuts – Past and future cones of an event –Models of process communications. Logical Time: A framework for a system of logical clocks –Scalar time –Vector time – Physical clock synchronization: NTP.

Module III: Message Ordering & Snapshots: [8L]

Message ordering and group communication: Message ordering paradigms – Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system – Group communication – Causal order (CO) - Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions – Snapshot algorithms for FIFO channels.

Module IV: Distributed Mutex & Deadlock : [8L]

Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport's algorithm – Ricart Agrawala algorithm – Maekawa's algorithm – Suzuki–Kasami's broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries – Models of deadlocks – Knapp's classification – Algorithms for the single resource model, the AND model and the OR model.

Module V: Recovery & Consensus: [8L]

Checkpointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated checkpointing algorithm – Algorithm for asynchronous checkpointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure – free system – Agreement in synchronous systems with failures.

Module VI: P2P & Distributed Shared Memory: [8L]

Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry. Distributed shared memory: Abstraction and advantages – Memory consistency models –Shared memory Mutual Exclusion.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Kshemkalyani, Ajay D., and Mukesh Singhal. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011.
2. George Coulouris, Jean Dollimore and Tim Kindberg, —Distributed Systems Concepts and Design, Fifth Edition, Pearson Education, 2012.

Reference Books:

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
2. Mukesh Singhal and Niranjan G. Shivaratri. Advanced concepts in operating systems. McGraw-Hill, Inc., 1994.
3. Tanenbaum A.S., Van Steen M., —Distributed Systems: Principles and Paradigms‖, Pearson Education, 2007.
4. Liu M.L., —Distributed Computing, Principles and Applications‖, Pearson Education, 2004.

Course Outcome (CO):

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

CO1: *Explain* the fundamental principles and execution models of distributed systems, including their relation to computer system components and parallel systems.

CO2: *Apply* logical and physical time mechanisms to analyze distributed executions and event ordering in communication networks.

CO3: *Analyze* message ordering guarantees, global state determination, and snapshot algorithms in asynchronous and FIFO–based distributed environments.

CO4: *Evaluate* distributed mutual exclusion and deadlock detection algorithms across different system models and resource allocation scenarios.

CO5: *Design* fault-tolerant distributed solutions by integrating checkpointing, recovery, consensus mechanisms, and distributed shared-memory or P2P overlays.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	2	2	-	-
CO2	2	-	2	2	-	-
CO3	3	2	3	3	2	-
CO4	3	-	3	3	2	2
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

DATA PREPARATION AND ANALYSIS

Course Information:

School: School of Science & Technologyg	**Course Type:** L-T-P
Name: Data Preparation and Analysis	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 1^{st}

Learning Objectives:

After completing this course, students will be able to:

1. Understand the fundamental concepts of Exploratory Data Analysis (EDA), identify different data types, and apply suitable tools and techniques for initial data exploration.
2. Perform essential data transformation tasks such as deduplication, discretization, binning, and handling missing data using traditional and statistical methods.
3. Analyze data using univariate, bivariate, and multivariate techniques, and work with time-series data through indexing, visualization, grouping, and resampling.
4. Learn basic predictive models such as linear regression, evaluate model performance, and interpret accuracy metrics; distinguish between supervised and reinforcement learning.
5. Learn statistical summarization techniques and create meaningful data visualizations using plots such as scatter plots, bar charts, and dot charts for effective interpretation.
6. Demonstrate clustering methods including spectral clustering, hierarchical clustering, EM-based clustering, and identify outliers through cluster analysis.
7. Implement dimensionality reduction techniques such as PCA, SVD, FA, MDS, manifold learning, and self-organizing maps to simplify high-dimensional datasets.

Prerequisite:

Students should have basic knowledge of statistics (mean, median, variance, probability), foundational data handling skills, and familiarity with programming using Python or R. Understanding of basic linear algebra (matrices, vectors), introductory data structures, and core concepts of supervised learning (like regression and classification) is also expected.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to Exploratory Data Analysis	8	16.66%
Module-II: Data Transformation	8	16.66%
Module-III: Correlation Analysis and Time Series	8	16.67%
Module-IV: Model Development, Data Summarization and Visualization	8	16.67%
Module-V: Clustering Algorithms	8	16.67%
Module-VI: Dimensionality Reduction	8	16.67%

Syllabus Outline:

Module I: Introduction to Exploratory Data Analysis: [8L]

Introduction to Exploratory Data Analysis (EDA) –Steps in EDA, Data Types: Numerical Data – Discrete data, continuous data – Categorical data – Measurement Scales: Nominal, Ordinal, Interval, Ratio – Comparing EDA with classical and Bayesian Analysis – Software tools for EDA.

Module II: Data Transformation: [8L]

Transformation Techniques: Performing data deduplication - replacing values – Discretization and binning. Introduction to Missing data, handling missing data: Traditional methods - Maximum Likelihood Estimation.

Module III: Correlation Analysis and Time Series: [8L]

Analysis Types of analysis: Univariate analysis - bivariate analysis - multivariate analysis. Time Series Analysis (TSA): Fundamentals of TSA - characteristics of TSA – Time based indexing -visualizing time series – grouping time series data - resampling time series data.

Module IV: Model Development, Data Summarization and Visualization: [8L]

Constructing linear regression model – evaluation – computing accuracy – understanding accuracy. Understanding reinforcement learning: Difference between supervised and reinforcement learning – Applications of reinforcement learning. Statistical summary measures, data elaboration, 1-D Statistical data analysis, 2-D Statistical data Analysis, contingency tables, n-D Statistical data analysis. Visualization: Scatter plots –Dot charts - Bar plots.

Module V: Clustering Algorithms: [8L]

Introduction to Spectral clustering – Document clustering – Minimum Spanning Tree clustering. Overview of Model-based clustering – Expectation-Maximization algorithm –Hierarchical Agglomerative model-based clustering. Outlier detection using Clustering.

Module VI: Dimensionality Reduction: [8L]

Linear Methods: Principal Component Analysis (PCA) – Singular Value Decomposition – Factor Analysis -Intrinsic Dimensionality. Non-Linear methods: Multidimensional Scaling – Manifold Learning – Self-Organizing Maps.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python" 1 st Edition, 2020, Packt Publishing.
2. Martinez, W , Martinez A & J.L. Solka : Exploratory Data Analysis with MATLAB, CRC Press, A Chapman & Hall Book, 3 rd Edition, 2017

Reference Books:

1. Michael Jambu, "Exploratory and multivariate data analysis", 1991, 1 st Edition, Academic Press Inc.
2. Charu C. Aggarwal, "Data Mining The Text book", 2015, Springer.
3. Craig K. Enders, "Applied Missing Data Analysis", 2010, 1 st Edition, The Guilford Press.

➠ Course Outcome (CO): .....

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

CO1: *Explain* the fundamental concepts of exploratory data analysis (EDA), data types, measurement scales, and EDA tools used for data understanding.

CO2: *Apply* data preprocessing techniques such as deduplication, binning, discretization, and missing-data handling to prepare datasets for analysis.

CO3: *Analyze* univariate, bivariate, multivariate patterns and time-series characteristics using statistical and visual analytical techniques.

CO4: *Evaluate* regression models, statistical summaries, reinforcement learning concepts, and data visualization methods for meaningful data interpretation.

CO5: *Design* and *implement* clustering, outlier detection, and dimensionality-reduction techniques (PCA, SVD, manifold learning) for high-dimensional data analysis.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	1	-	1	-
CO2	2	2	2	2	2	-
CO3	3	2	2	2	2	-
CO4	3	3	2	2	2	1
CO5	3	3	3	3	3	2

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

RESEARCH METHODOLOGY AND IPR

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Research Methodology and IPR	**Course Credit:** 02[2-0-0]
Department: Computer Science Engineering	**Category:** SEC/PSE
Code: XXXXXXXXX	**Semester:** 1^{st}

Learning Objectives: After completing this course, students will be able to:

1. Understand the fundamentals of research methodology, problem identification, and formulation.
2. Able to conduct structured literature reviews, identify research gaps, and write ethical, plagiarism-free research documents.
3. Acquire knowledge of various forms of Intellectual Property Rights (IPR) such as patents, copyrights, trademarks, and their relevance in technological innovation.
4. Understand international procedures, treaties, and frameworks governing IPR protection and patent filing.
5. Know about patent systems, licensing, technology transfer, and emerging IPR issues to real-world research and innovation scenarios.

Prerequisite:

Students should have basic understanding of scientific writing and technical communication, familiarity with computer science or engineering research domains, ability to analyze academic documents and interpret technical information, foundational understanding of innovation processes, technology development, and legal/ethical practices in computing.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	16.66%
Module-II: Literature studies and Plagiarism	8	16.66%
Module-III: Nature of Intellectual Property	8	16.67%
Module-IV: International Scenario	8	16.67%
Module-V: Patent Rights	8	16.67%
Module-VI: New Developments in IPR	8	16.67%

Syllabus Outline:

Module I: Introduction: [8L]

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations.

Module II: Literature studies and Plagiarism: [8L]

Effective literature studies approaches, analysis Plagiarism, Research ethics, Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee.

Module III: Nature of Intellectual Property: [8L]

Patents, Designs, Trade and Copyright. Process of Patenting and Development: technological research, innovation, patenting, development.

Module IV: International Scenario: [8L]

International cooperation on Intellectual Property. Procedure for grants of patents, Patenting under PCT.

Module V: Patent Rights: [8L]

Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications.

Module VI: New Developments in IPR: [8L]

Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Stuart Melville and Wayne Goddard, “Research methodology: an introduction for science & engineering students”.
2. Wayne Goddard and Stuart Melville, “Research Methodology: An Introduction”.

3. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners".

Reference Books:

1. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd, 2007.
2. Mayall, "Industrial Design", McGraw Hill, 1992.
3. Niebel, "Product Design", McGraw Hill, 1974.
4. Asimov, "Introduction to Design", Prentice Hall, 1962.
5. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
6. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008

Course Outcome (CO):

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

CO1: *Identify* and *define* appropriate research problems, explain research objectives, and select suitable data collection and analysis methods in alignment with research fundamentals.

CO2: *Apply* effective literature survey strategies, evaluate research ethics, plagiarism checks, and produce technically sound academic documents such as reports, proposals, and presentations.

CO3: *Analyze* various forms of Intellectual Property Rights (IPR) including patents, copyrights, designs, and trademarks, and evaluate their role in innovation

CO4: *Evaluate* international IPR frameworks, including PCT procedures, and apply relevant guidelines for patent filing and protection at the global level.

CO5: *Interpret* and *assess* patent rights, licensing mechanisms, technology transfer processes, and emerging developments in IPR through real-world case studies.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	1
CO2	2	3	1	-	-	2
CO3	2	2	2	1	-	2
CO4	2	3	2	-	-	2
CO5	2	3	2	1	-	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

OBJECT ORIENTED ANALYSIS AND DESIGN LAB

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P-S
Name: Object Oriented Analysis and Design Lab	**Course Credit:** 02[0-0-4-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 1^{st}

List of Practicals:

1. Identify a software system that needs to be developed
2. Document the Software Requirements Specification (SRS) for the identified system
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.
5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
10. Implement the modified system and test it for various scenarios

ADVANCED DATA STRUCTURES LAB

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P-S
Name: Advanced Data Structures Lab	**Course Credit:** 02[0-0-4-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 1st

List of Practicals:

1. Implement a stack by using array then do the PUSH & POP operation
2. Write a program to evaluate a postfix notation.
3. Write a program to convert infix to postfix.
4. Implement a Circular Queue by using array then do the enqueue and dqueue operation.
5. Implement Single Linked List and does insertion, deletion, display, reverse.
6. Implement Doubly Linked List and does insertion, deletion, display, reverse.
7. Implement a stack using linked lists.
8. Implement Circular Linked List, queue using linked lists.
9. Implement JOSEPHUS problem.
10. Write a program to add two polynomials.
11. Write a program to multiply two polynomials.
12. Write a program for addition of sparse matrix.
13. Write a program to multiplication of sparse matrix.
14. Create binary search tree and implement Preorder, Inorder, Postorder and delete an element from the tree
15. Implement a threaded binary tree and perform the inorder traversal operation.
16. Implement AVL tree.
17. Implement Splay tree.
18. Implement Priority Queue using Heap.
19. Implement BFS, DFS.
20. Implement Prim's and Kruskal's Algorithm.
21. Write a program to sort an array using Bubble sort.
22. Write a program to sort an array using Insertion sort
23. Write a program to sort an array using Selection sort.
24. Write a program to sort an array using Quick sort.
25. Write a program to sort an array using Merge sort.
26. Write a program to sort an array using Heap sort.

27. Write a program to sort an array using Radix sort.
28. Write a program to sort an array using Shell sort.
29. Implement Linear and Binary search.
30. Implement interpolation search.

SEMESTER-II

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	High Performance Computing		CC/PCC	4	3	1	0	0
2	Advanced Software Engineering/ Web Analytics and Development		DSE/PEC	4	3	1	0	0
3	Fog and Edge Computing/ Soft Computing		DSE/PEC	4	3	1	0	0
4	Machine Learning/ Quantum Computing		DSE/PEC	4	3	1	0	0
5	Foreign Language II		USC/MUS	2	2	0	0	0
6	Mini Project with Seminar / Industry Research Project		PROJECT	3	0	0	0	6
Practical								
7	Machine Learning Lab/ Quantum Computing Lab		DSE/PEC	1	0	0	2	0
			Total Credit:	22				

HIGH PERFORMANCE COMPUTING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: High Performance Computing	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives:

After completing this course, students will be able to:

1. Provide knowledge on high performance computing concepts to the students.
2. Comprehend the students how to analyze the parallel programming through OpenMP, MPI, CUDA.
3. Teach the student how to apply job management techniques and evaluate the performance.

Prerequisite:

Students should have prior knowledge of operating systems (process management, scheduling, memory management), computer architecture (CPU, memory hierarchy, parallelism), and proficiency in programming using C/C++ or Python. Basic understanding of algorithms, data structures, and introductory parallel programming concepts (threads, processes, concurrency) is also required. Familiarity with linear algebra and numerical computation is recommended.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to High Performance Computing (HPC)	8	16.66%
Module-II: HPC Paradigms	8	16.66%
Module-III: Parallel Programming	8	16.67%
Module-IV: Job Management Systems	8	16.67%
Module-V: Achieving Performance	8	16.67%
Module-VI: HPC Benchmarks	8	16.67%

Syllabus Outline:

Module I: Introduction to High Performance Computing (HPC): [8L]
Overview of Parallel Computers and high-performance computing (HPC), History of HPC, Numerical and HPC libraries, Performance metrics.

Module II: HPC Paradigms: [8L]
Supercomputing, Cluster Computing, Grid Computing, Cloud Computing, many core Computing, Petascale Systems.

Module III: Parallel Programming: [8L]
Introduction to OpenMP, Parallel constructs, Runtime Library routines, Work-sharing constructs, Scheduling clauses, Data environment clauses, atomic, master Nowait Clause, Barrier Construct, overview of MPI, MPI Constructs, OpenMP vs MPI. Introduction to GPU Computing, CUDA Programming Model, CUDA API, Simple Matrix, Multiplication in CUDA, CUDA Memory Model, Shared Memory Matrix Multiplication, Additional CUDA API Features.

Module IV: Job Management Systems: [8L]
Batch scheduling: Condor, Slurm, SGE, PBS, Light weight Task Scheduling: Falkon, Sparrow.

Module V: Achieving Performance: [8L]
Measuring performance, identifying performance bottlenecks, Partitioning applications for heterogeneous resources, Using existing libraries and frameworks.

Module VI: HPC Benchmarks: [8L]
HTC, MTC (Many Tasks Computing), Top 500 Super computers in the world, Top 10 Super Computer architectural details, Exploring HPC Benchmarks: HPL, Stream

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Victor Eijkhout, Edmond Chow, Robert van de Geijn, Introduction to High Performance Scientific Computing, 2nd edition, revision 2016.
2. Rob Farber, CUDA Application Design and Development, Morgan Kaufmann Publishers, 2013.

Reference Books:

1. Zbigniew J. Czech, Introduction to parallel computing, 2nd edition, Cambridge University Press,2016.

Course Outcome (CO):

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

CO1: *Understand* and analyze the overview and performance metrics of high-performance computing.

CO2: *Comprehend* the various high-performance computing paradigms and job management systems.

CO3: *Design* and *develop* applications using OpenMP, MPI, and CUDA, and apply high-performance computing concepts to solve real-world computational problems.

CO4: *Analyze* benchmarks of high-performance computing systems.

CO5: *Demonstrate* emerging trends and technologies in high-performance computing.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	2	3	1	2
CO2	2	1	2	3	1	2
CO3	3	2	3	3	2	3
CO4	3	2	3	3	1	3
CO5	2	1	3	3	2	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ADVANCED SOFTWARE ENGINEERING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Advanced Software Engineering	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives:

After completing this course, students will be able to:

1. To introduce the essential software engineering concepts involved.
2. To impart skills in the design and implementation of efficient software systems across disciplines.
3. To familiarize engineering practices and standards used in developing software products and components.

Prerequisite:

Students should have Fundamental knowledge of Software Engineering, including software development lifecycle (SDLC), requirements, design, coding, and testing; Understanding of object-oriented programming (Java/C++/Python) and basic design principles, Familiarity with data structures and algorithms, enabling structured problem-solving; Basic knowledge of UML modeling, documentation, and version control systems (Git); Exposure to software project management concepts, such as effort estimation and scheduling, is helpful; Optional but beneficial: Experience with team-based software development or software engineering lab/project.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Overview of Software Engineering and Software	8	16.66%
Module-II: Modelling Requirements	8	16.66%
Module-III: Software Design	8	16.67%
Module-IV: Validation and Verification	8	16.67%
Module-V: Software Evolution	8	16.67%
Module-VI: Quality Assurance	8	16.67%

Syllabus Outline:

Module I: Overview of Software Engineering and Software: [8L]

Nature of Software, Software Engineering, Software process, project, product, Process Models, Classical Evolutionary models, Overview of System Engineering, Planning scope, milestones deliverables, Risk Management, Metrics Measurement.

Module II: Modelling Requirements: [8L]

Requirements Engineering process Requirement Elicitation, System Modelling - Requirements Specification and Requirement Validation.

Module III: Software Design: [8L]

Design concepts and principles - Abstraction - Refinement - Modularity Cohesion coupling, Architectural design, Detailed Design Transaction Transformation, Refactoring of designs, Object-oriented Design User-Interface Design.

Module IV: Validation and Verification: [8L]

Strategic Approach to Software Testing, Testing Fundamentals Test Plan, Test Design, Test Execution, Reviews, Inspection Auditing.

Module V: Software Evolution: [8L]

Software Maintenance, Types of Maintenance, Software Configuration Management, Overview of RE-engineering Reverse Engineering.

Module VI: Quality Assurance: [8L]

Introduction to Software Quality: Definition of software quality, Quality factors and quality attributes (McCall, Boehm models), Dimensions of product and process quality. Software Quality Assurance (SQA) Framework: SQA activities and goals, SQA architecture, organizational structure, SQA plan and its components. Quality Standards and Models: ISO 9001 for software, Capability Maturity Model (CMM and CMMI), Six Sigma concepts for software, IEEE software quality standards. Software Reviews and Audits: Technical reviews, walkthroughs, inspections; Formal vs informal reviews, Audit process, types of audits, compliance checking, Quality Metrics, Quality Control Techniques, Software Reliability & Safety, Basic concepts of software reliability, Continuous Quality Improvement.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Roger Pressman, Software Engineering: A Practitioner"s Approach, 7th Edition, McGraw-Hill, 2010

Reference Books:

1. Ian Sommerville, Software Engineering, 9th Edition, Addision-Wesley, 2016
2. Pankaj Jalote, A Concise Introduction to Software Engineering, Springer, 2008
3. William E. Lewis , Software Testing and Continuous Quality Improvement, Third Edition, Auerbach Publications, 2008

Course Outcome (CO):

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

CO1: *Explain* and *Compare* various software process models and fundamental software engineering principles, including risk management and quality assurance techniques.

CO2: *Analyze* and *Validate* complex software requirements through elicitation, specification, and system modeling.

CO3: *Design* and *Develop* software architecture and detailed designs by applying principles of abstraction, modularity, cohesion, and coupling.

CO4: *Evaluate* the quality of a software product using appropriate strategic testing, reviews, and inspection methods.

CO5: *Formulate* strategies for software evolution, including configuration management, maintenance, and re-engineering processes.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	3	1	1	2	3	2
CO2	2	3	-	3	3	2
CO3	3	3	3	2	3	3
CO4	2	3	-	1	3	3
CO5	2	2	1	2	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

WEB ANALYTICS AND DEVELOPMENT

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Web Analytics and Development	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives: After completing this course, students will be able to:

1. Understand key concepts, processes, and the need for web analytics.
2. Apply common metrics like bounce rates and page views to optimize website performance.
3. Learn techniques for gathering and interpreting quantitative and qualitative data.
4. Gain insights into Web Analytics 2.0, Google Analytics, and digital marketing optimization.

Prerequisite:

Students should have basic knowledge of web technologies (HTML, HTTP, client–server architecture), understanding of data analytics, descriptive statistics, and data visualization; familiarity with web development concepts, cookies, sessions, and tracking mechanisms; foundation in Python/R or any analytical tool for processing web datasets, understanding of database systems and structured/unstructured data.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	10%
Module-II: Data Collection	8	20%
Module-III: Qualitative Analysis	8	20%
Module-IV: Web Analytic fundamentals	8	20%
Module-V: Web Metrics and Common metrics	8	10%
Module-VI: Web analytics 2.0 and Google Analytics	8	20%

Syllabus Outline: .....

Module I: Introduction: [8L]

Definition, Process, Key terms: Site references, Keywords and Key phrases; building block terms: Visit characterization terms, Content characterization terms, Conversion metrics; Categories: Offsite web, On site web; Web analytics platform, Web analytics evolution, Need for web analytics, Advantages, Limitations.

Module II: Data Collection: [8L]

Clickstream Data: Web logs, Web Beacons, JavaScript tags, Packet Sniffing; Outcomes Data: E-commerce, Lead generation, Brand/Advocacy and Support; Research data: Mindset, Organizational structure, Timing; Competitive Data: Panel-Based measurement, ISP-based measurement, Search Engine data.

Module III: Qualitative Analysis: [8L]

Heuristic evaluations: Conducting a heuristic evaluation, Benefits of heuristic evaluations; Site Visits: Conducting a site visit, Benefits of site visits; Surveys: Website surveys, Post-visit surveys, Creating and running a survey, Benefits of surveys.

Module IV: Web Analytic fundamentals: [8L]

Capturing data: Web logs or JavaScripts tags, Separate data serving and data capture, Type and size of data, Innovation, Integration, selecting optimal web analytic tool, Understanding clickstream data quality, Identifying unique page definition, Using cookies, Link coding issues.

Module V: Web Metrics and Common metrics: [8L]

Hits, Page views, Visits, Unique visitors, Unique page views, Bounce, Bounce rate, Page/visit, Average time on site, New visits; Optimization (e-commerce, non-e-commerce sites): Improving bounce rates, Optimizing adwords campaigns; Real time report, Audience report, Traffic source report, Custom campaigns, Content report, Google analytics, Introduction to KPI, characteristics, Need for KPI, Perspective of KPI, Uses of KPI.

Module VI: Web analytics 2.0 and Google Analytics: [8L]

Web analytics 1.0, Limitations of web analytics 1.0, Introduction to analytic 2.0, Competitive intelligence analysis: CI data sources, Toolbar data, Panel data ,ISP data, Search engine data, Hybrid data, Website traffic analysis: Comparing long term traffic trends, Analyzing competitive site overlap and opportunities. Brief introduction and working, Adwords, Benchmarking, Categories of traffic: Organic traffic, Paid traffic; Google website optimizer, Implementation technology, Limitations, Performance concerns, Privacy issues.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)

- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Clifton B., Advanced Web Metrics with Google Analytics, Wiley Publishing, Inc. (2010), 2nd ed.
2. Kaushik A., Web Analytics 2.0 The Art of Online Accountability and Science of Customer Centricity, Wiley Publishing, Inc. (2010),1st ed.

Reference Books:

1. Sterne J., Web Metrics:Proven methods for measuring web site success, John Wiley and Sons (2002),1sted

Course Outcome (CO):

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

CO1: *Explain* the foundational concepts of web analytics, key metrics, analytics evolution, and the need for data-driven website optimization.

CO2: *Apply* different web data collection techniques (clickstream, competitive, outcomes, and research data) for analyzing user behavior and site performance.

CO3: *Analyze* qualitative evaluation techniques such as heuristic reviews, surveys, and site visits to assess user experience and website usability.

CO4: *Evaluate* web analytic tools, clickstream data quality, page definitions, cookie usage, and link coding issues to ensure accurate data capture and reporting.

CO5: *Design* and *interpret* advanced web analytics solutions using KPIs, Google Analytics, optimization metrics, competitive intelligence frameworks, and Web Analytics 2.0 methodologies.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	-	2	-
CO2	2	-	3	2	3	-
CO3	2	2	2	-	2	-
CO4	3	2	3	2	2	1
CO5	3	3	3	2	3	2

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

FOG AND EDGE COMPUTING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Fog and Edge Computing	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives: After completing this course, students will be able to:

1. Introduce cloud computing and enabling technologies
2. Explore the need for fog and edge computation
3. Impart the knowledge to log the sensor data and to perform further data analytics

Prerequisite: Principles of Cloud Computing

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Internet of Things (IoT) and New Computing Paradigms	4	8%
Module-II: Challenges in Federating Edge Resources	6	12%
Module-III: Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds	10	21%
Module-IV: Optimization Problems in Fog and Edge Computing	10	21%
Module-V: Middleware for Fog and Edge Computing and Technologies in Fog Computing	10	21%
Module-VI: Applications and Issues	8	17%

Syllabus Outline: .....

Module I: Internet of Things (IoT) and New Computing Paradigms: [4L]

Introduction-Relevant Technologies-Fog and Edge Computing Completing the Cloud-Hierarchy of Fog and Edge Computing-Business Models-Opportunities and Challenges.

Module II: Challenges in Federating Edge Resources: [6L]

Introduction-Methodology-Integrated C2F2T Literature by Modeling Technique-Integrated C2F2T

Literature by Use-Case Scenarios-Integrated C2F2T Literature by Metrics-Future Research Directions.

Module III: Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds: [10L]

Introduction-Background-Network Slicing-Network Slicing in Software-Defined Clouds-Network Slicing Management in Edge and Fog- Internet of Vehicles: Architecture, Protocol and Security Seven layered model architecture for Internet of Vehicles- IoV: Network Models, Challenges and future aspects.

Module IV: Optimization Problems in Fog and Edge Computing: [10L]

Preliminaries-The Case for Optimization in Fog Computing-Formal Modeling Framework for Fog Computing-Metrics-Further Quality Attributes-Optimization Opportunities along the Fog Architecture-Optimization Opportunities along the Service Life Cycle-Toward a Taxonomy of Optimization Problems in Fog Computing.

Module V: Middleware for Fog and Edge Computing and Technologies in Fog Computing: [10L]

Need for Fog and Edge Computing Middleware-Design Goals-State-of-the-Art Middleware Infrastructures-System Model-Proposed Architecture-Case Study Example. Fog Data Management-Motivating Example: Smart Building-Predictive Analysis with FogTorch-Machine Learning in Fog Computing-Data Analytics in the Fog-Data Analytics in the Fog-Architecture-Configurations.

Module VI: Applications and Issues: [8L]

Exploiting Fog Computing in Health Monitoring-Smart Surveillance Video Stream Processing at the Edge for Real-Time Human Objects Tracking-Fog Computing Model for Evolving Smart Transportation Applications-Testing Perspectives of Fog-Based IoT Applications-Legal Aspects of Operating IoT Applications in the Fog.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Buyya, Rajkumar, and Satish Narayana Srirama, eds, Fog and edge computing: principles and paradigms, 1st edition, John Wiley & Sons, 2019.
2. John Mutumba Bilay , Peter Gutsche, Mandy Krimmel and Volker Stiehl ,SAP Cloud Platform Integration: The Comprehensive Guide, 2nd edition, Rheinwerg publishing, 2019

Reference Books:

1. Bahga, Arshdeep, and Vijay Madisetti. Cloud computing: A hands-on approach, 1st edition, CreateSpace Independent Publishing Platform, 2013.
2. Ovidiu Vermesan, Peter Friess, Internet of Things –From Research and Innovation to Market Deployment, 1 stedition,River Publishers, 2014
3. Michael Missbach, Thorsten Staerk, Cameron Gardiner, Joshua McCloud, Robert Madl, Mark Tempes, George Anderson, SAP on Cloud, 1 st edition, Springer, 2016

Course Outcome (CO):

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

CO1: *Explain* the fundamental concepts and hierarchy of Fog and Edge Computing by relating them to the limitations and capabilities of Cloud computing.

CO2: *Illustrate* the mechanisms of network slicing management in 5G, Fog, Edge, and Clouds with a focus on the Internet of Vehicles (IoV) architecture and protocols.

CO3: *Differentiate* the various optimization opportunities and quality attributes in the Fog Computing architecture to propose metrics for performance evaluation.

CO4: *Critique* the design goals and state-of-the-art middleware infrastructures for Fog and Edge Computing based on their suitability for smart application development.

CO5: *Develop* a conceptual Fog-based IoT application model (e.g., for health monitoring or smart transportation) incorporating data analytics and testing perspectives.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	1	1	2	1	2	1
CO2	2	2	3	2	3	2
CO3	3	2	3	3	3	2
CO4	3	3	3	3	3	3
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

SOFT COMPUTING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Soft Computing	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives:

On completion of the course, student will be able to: Demonstrate artificial intelligence in terms of linguistic variable concepts related to design of modern AI tools in several domain including healthcare, finance, agriculture etc. Analyse the performance of AI tools with data availability. This course is intended to teach the basics application in AI application.

Prerequisite:

A strong mathematical background- Proficiency with algorithm set theory, mathematical logic, Programming skills python, Perl, MATLAB, etc. and critical thinking and problem-solving skills.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	2	4%
Module-II: Fuzzy Sets	8	17%
Module-III: Fuzzy Systems	12	25%
Module-IV: Artificial Neural Network	14	29%
Module-V: Genetic Algorithm	6	12.5%
Module-VI: Associated Soft Computing Techniques	6	12.5%

Syllabus Outline:

Module I: Introduction: [2L]

Introduction to soft computing, requirement, soft computing versus hard computing, different tool and techniques and applications, Computational Intelligence versus Machine Learning basics.

Module II: Fuzzy Sets: [8L]

Introduction, Fuzzy sets versus crisp sets, operations on Classical sets, properties of classical sets, Fuzzy set operations, properties of fuzzy sets, cardinality, operations, Fuzzy relations and properties of fuzzy relations.

Module III: Fuzzy Systems: [12L]

Membership functions: Features of membership functions, standard forms and boundaries, fuzzification, for fuzzy sets, Defuzzification methods: Lamba Cuts, Alpha cuts Fuzzy Logic, Approximate reasoning and Fuzzy Implication. Fuzzy Rule based Systems: Linguistic Hedges, Fuzzy Rule based system – Aggregation of fuzzy Rules, Fuzzy Inference System- Mamdani Fuzzy Models – Sugeno Fuzzy Models. Applications of Fuzzy Logic, fuzzy logic controllers, fuzzy pattern recognition, fuzzy image processing.

Module IV: Artificial Neural Network: [14L]

Introduction and basic models, biological neurons and artificial neural network, Learning Methods: Mc-pitt , Hebb's learning, Perceptron, Adaline and Madaline networks, single layer network, Multilayer feed forward network, Back-propagation network, Different issue regarding convergence multilayer perceptron, Competitive learning, Self-Organizing Maps, Hopfield Networks, Associative Memories, Boltzmann Machine and applications.

Module V: Genetic Algorithm: [6L]

Introduction, different operators of GA: crossover and mutation, analysis of selection operations, Hypothesis and building block, Multi-objective Genetic Algorithm (MOGA), GA in search and optimization and applications.

Module VI: Associated Soft Computing Techniques: [6L]

Simulated Annealing, Tabu search, Ant colony optimization (ACO), Particle Swarm Optimization (PSO), Hybrid Systems: Neural Network based Fuzzy system, Fuzzy Logic based Neural Networks.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. 1. Jang, Jyh-Shing Roger. "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence." Prentice Hall, 1997.

2. Bezdek, James C., and Sankar K. Pal. "Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data." IEEE Press, 1992.
3. Jain, Lakhmi C., et al. "Neuro-Fuzzy and Soft Computing: A Computational Approach." CRC Press, 2017.

Reference Books:

1. Kosko, Bart. "Fuzzy Thinking: The New Science of Fuzzy Logic." Hyperion, 1993.
2. Pedrycz, Witold, and Fernando Gomide. "An Introduction to Fuzzy Sets: Analysis and Design." MIT Press, 1998.

Course Outcome (CO):

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

CO1: *Explain* the fundamental concepts, need, and applications of soft computing, and distinguish it from hard computing and conventional AI/ML techniques.

CO2: *Apply* fuzzy set theory, fuzzy relations, and fuzzy operations for solving real-world problems involving uncertainty and imprecision.

CO3: *Analyze* fuzzy inference systems—including Mamdani and Sugeno models—to design fuzzy rule–based systems for decision-making and pattern recognition applications.

CO4: *Evaluate* neural network learning models (Perceptron, MLP, SOM, Hopfield, Boltzmann Machine) and compare their performance across different computational intelligence tasks.

CO5: *Design* hybrid and evolutionary soft computing solutions using Genetic Algorithms, PSO, ACO, Simulated Annealing, and neuro-fuzzy architectures for optimization and complex problem-solving.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	-	2	-
CO2	2	-	3	-	3	-
CO3	2	1	3	-	3	1
CO4	3	2	3	2	3	2
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

MACHINE LEARNING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Machine Learning	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives: After completing this course, students will be able to:

1. Understand the basic theory underlying machine learning.
2. Able to formulate machine learning problems corresponding to different applications.
3. Understand a range of machine learning algorithms along with their strengths and weaknesses.
4. Able to apply machine learning algorithms to solve problems of moderate complexity.
5. Apply the algorithms to a real-world problem, optimize the models learned, and report on the expected accuracy that can be achieved by applying the models.

Prerequisite:

Knowledge of Artificial Intelligence, Linear algebra, Calculus, Mathematical logic and differential equation.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to Machine Learning	4	8%
Module-II: Feature Engineering	6	13%
Module-III: Classification	14	29%
Module-IV: Clustering	12	25%
Module-V: Machine Learning System Design	8	17%
Module-VI: Case studies	4	8%

Syllabus Outline: .....

Module I: Introduction to Machine Learning: [4L]

Basic Concepts of Machine Learning, Types of Machine Learning, Supervised Learning Versus Unsupervised Learning Versus Reinforcement Learning, Discriminative Algorithms.

Module II: Feature Engineering: [6L]

Introduction to Data Processing, ETL, Measurement of Purity, Entropy and Gini Index, Normalization and Standardization, Dimension Reduction, ICA (Independent Components Analysis), EM. Mixture of Gaussians, Factor Analysis, Normal Distribution and Gaussian Distribution.

Module III: Classification: [14L]

Introduction to Supervised Learning, Concepts of Linear Algebra, Linear Regression and Logistic Regression, Concepts Bias/ Variance Trade off, Prediction Versus Classification Problem, Naive Bayes, Maximum Entropy, Perceptron, Basic Concept of Neural Network, Generative Learning Algorithms, Gradient Descent, Regularization, Feed Forward Neural Network,Back Propagation Neural Network, Gaussian Discriminant Analysis, Conceptsof vectorization , Support Vector Machines, Introduction of Deep Learning, Hidden Markov Model, Genetic Algorithms,

Module IV: Clustering: [12L]

Introduction to Unsupervised learning: Introduction to Clustering, K-means and Hierarchical Clustering, Comparison among classification and clustering, Dimension reduction: PCA (Principal Components Analysis), Factor analysis.

Module V: Machine Learning System Design: [8L]

Underfitting and Overfitting Problem, Bias-Variance as Function of Lambda, Cross Validation, Learning Curves, Error Analysis, Confusion Matrix, Trading off Precision and Recall, ROC Curve, F1-Score and Accuracy Analysis.

Module VI: Case studies: [4L]

Applications of MLin Case Studies.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Bishop, Christopher M.Pattern Recognitionand MachineLearning.Springer,2006.
2. Alpaydin, Ethem.Introduction to Machine Learning.3rd ed.,The MIT Press, 2014.
3. Murphy, Kevin P.Machine Learning:A Probabilistic Perspective.The MIT Press, 2012.
4. Marsland, Stephen.Machine Learning:An Algorithmic Perspective.2nd ed.,CRC Press, 2014.

Reference Books:

1. Shalev-Shwartz, Shai, and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.
2. Géron, Aurélien.Hands-On Machine Learning with Scikit-Learn, Keras,and TensorFlow. 2nd ed., O'Reilly Media, 2019.
3. Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer, 2009.

Course Outcome (CO):

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

CO1: *Define* and *explain* the fundamental concepts, types, and principles of Machine Learning, including supervised, unsupervised, and reinforcement learning.

CO2: *Apply* feature engineering techniques for model ready datasets.

CO3: *Analyze* various supervised and unsupervised algorithms to determine their suitability for different prediction tasks.

CO4: *Evaluate* several machine learning models using different performance metrics and understand the datasets for fine tuning.

CO5: *Design* and *develop* complete ML system pipelines and implement case-study–based solutions using appropriate algorithms and performance-optimization strategies

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	1	1	2	1	2	1
CO2	2	1	2	1	2	3
CO3	3	2	3	3	2	2
CO4	3	2	3	3	1	3
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

QUANTUM COMPUTING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Quantum Computing	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

Learning Objectives:

After completing this course, students will be able to:

1. To understand the fundamental concepts on quantum computing
2. To learn how to do computation using quantum algorithms
3. To process secure information in various modern-day applications

Prerequisite:

Students should possess Strong foundation in linear algebra (vectors, matrices, eigenvalues, tensor products), good understanding of probability theory and complex numbers, prior exposure to classical algorithms and computational complexity, basic knowledge of cryptography, especially symmetric and public-key systems; familiarity with Python/Mathematical computing tools (NumPy, Qiskit optional but helpful).

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to Quantum Information	8	16%
Module-II: Quantum Algorithms Basic	8	17%
Module-III: Quantum Algorithms Advanced	8	17%
Module-IV: Quantum True Random Number Generators (QTRNG)	8	17%
Module-V: Basic Quantum key distribution	6	12%
Module-VI: Advanced Quantum key distribution	10	21%

Syllabus Outline:

Module I: Introduction to Quantum Information: [8L]
States, Operators, Measurements, Quantum Entanglement: Quantum Teleportation, Super-dense coding, CHSH Game, Quantum gates and circuits.

Module II: Quantum Algorithms Basic: [8L]
Deutsch-Jozsa, Simon, Grover, Shor, Implication of Grover's and Simon's algorithms towards classical symmetric key cryptosystems.

Module III: Quantum Algorithms Advanced: [8L]
Implication of Shor's algorithm towards factorization and Discrete Logarithm based classical public key cryptosystems.

Module IV: Quantum True Random Number Generators (QTRNG): [8L]
Quantum True Random Number Generators (QTRNG): Detailed design and issues of quantumness, Commercial products and applications.

Module V: Basic Quantum key distribution: [6L]
Quantum key distribution (QKD): BB84, Ekert, Semi-Quantum QKD protocols.

Module VI: Advanced Quantum key distribution: [10L]
Variations in Semi-Quantum QKD protocols, Issuesof Device Independence, Commercial products.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press. 2010.
2. Chris Bernhardt, Quantum Computing for Everyone, MIT Press 2019.

Reference Books:

1. Presskil Lecture notes: Available online: http://www.theory.caltech.edu/ preskill/ph229/
2. NIST Post Quantum Cryptography, Available online: https://csrc.nist.gov/projects/post-quantum

cryptography

Course Outcome (CO):

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

CO1: *Explain* fundamental quantum information concepts including qubit states, operators, measurements, entanglement, and quantum gates.

CO2: *Apply* basic quantum algorithms such as Deutsch–Jozsa, Simon, Grover, and Shor to analyze computational advantages over classical systems.

CO3: *Analyze* the security implications of quantum algorithms on classical cryptographic schemes such as symmetric key systems and public-key systems.

CO4: *Evaluate* the architecture, functioning, and quantumness of Quantum True Random Number Generators (QTRNG) and their real-world applications.

CO5: *Design* and *assess* basic and advanced Quantum Key Distribution (QKD) protocols, including BB84, Ekert, and Semi-Quantum QKD systems, with device-independence considerations.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	2	-	2	-
CO2	3	1	3	-	3	1
CO3	3	1	3	-	3	2
CO4	2	2	2	2	2	-
CO5	3	2	3	2	3	2

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

MACHINE LEARNING LAB

Course Information:

School: School of Engineering	**Course Type:** L-T-P
Name: Machine Learning Lab	**Course Credit:** 01[0-0-2]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

List of Practicals:

1. Write a Program to perform the following operations on matrices: a) Matrix addition
   a) Matrix Subtraction
   b) Matrix Multiplication
   c) Matrix Inversion
   d) Transpose of a Matrix
2. Write a Program to perform the following operations: a) Find the minimum and maximum element of the matrix
   a) Find the minimum and maximum element of each row in the matrix
   b) Find the minimum and maximum element of each column in the matrix
   c) Find trace of the given matrix
   d) Find rank of the given matrix
   e) Find eigenvalues and eigenvectors of the given matrix
3. Write a Program to find the mean, median, standard deviation, and mode using user-defined functions.
4. Create a data frame with columns and at least 5 observations:
   a) Retrieve a particular column from the DataFrame
   b) Summarize the data frame and observe the statistics of the DataFrame created
   c) Observe the mean and standard deviation of the data frame and print the values.
5. Write a program to implement Linear Regression for a sample training dataset stored as a .CSV file. Compute Mean Square Error by considering a few test data sets.
6.
7. Write a program to implement Non-linear Regression for a sample training dataset stored as a .CSV file. Compute Mean Square Error by considering a few test data sets.
8. Write a program to implement Logistic Regression for a sample training dataset stored as a .CSV file. Compute the accuracy of the classifier.
9. Write a program to implement the naïve Bayesian classifier for a sample training dataset stored as a .CSV file. Compute the accuracy of the classifier, considering a few test datasets.

10. Write a program to implement the k-Nearest Neighbor algorithm to classify the iris dataset. Print both correct and wrong predictions.
11. Write a program to implement the Support Vector Machine algorithm to classify the iris dataset. Print both correct and wrong predictions.
12. Write a program to demonstrate the working of the decision tree based on the ID3 algorithm. Use an appropriate dataset for building the decision tree and apply this knowledge to classify a new sample.
13. Write a program to demonstrate the working of the decision tree based on the CART algorithm. Use an appropriate dataset for building the decision tree and apply this knowledge to classify a new sample.
14. Write a program to construct a Regression tree for cost estimation by assuming any numerical dataset.
15. Write a program to calculate the accuracy, precision, and recall for your dataset. Assume a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task.
16. Implement a single neural network and test for different logic gates.
17. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate datasets.

QUANTUM COMPUTING LAB

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Quantum Computing Lab	**Course Credit:** 01[0-0-2]
Department: Computer Science Engineering	**Category:** DSE/PEC
Code: XXXXXX	**Semester:** 2^{nd}

List of Practicals:

Section 1

Working with Python in Colab Open your Google Account here https://colab.research.google.com (Refer the video for additional functionalities of Colab https://www.youtube.com/watch?v=rNgswRZ2C1Y) Students can refer to https://www.pythonpool.com/python-vector/ or any other website to learn how to use vectors in Python.

- Import NumPy and math package.
- Create variables by considering $a = 1/\sqrt{2}$ and $b = 1/\sqrt{2}$
- Create a vector or One-dimensional array using variables a and b.
- Check whether the vector is normalized or not. Vector is normalized if a $2+b2 = 1$ in this particular case.
- Find the length of a vector.

Section 2

Vector in Python Create a new notebook and add the following functionalities in the Colab note-book:

- Create a vector and read the elements of the vector from the keyboard.
- Write a function that will check whether the function is normalized or not.
- Write a function to find the dot and cross product of two vectors (A and B) using NumPy.
- Write a function to find a unit vector corresponding to vector A.

Section 3

Matrices in Python Create a new notebook and add the following functionalities in the Colab notebook using NumPy:

- Read a matrix from the Keyboard.
- Find the transpose of a matrix.
- Find the Determinant of a matrix (Usinglinalg.det in NumPy).
- Find Rank of a matrix (Usinglinalg.matrixrank in numpy).

- Find Trace of the matrix
- Find Eigenvalue and Eigenvectors of a square matrix (Use linalg.eig).
- Find the Inverse of a matrix. 1

Section 4

Complex Number in Python Refer to https://realpython.com/python-complex-numbers/using-python-complex-numbers-as-2d-vectors or any other site to read how to use complex numbers in python. for Create a new notebook and add the following functionalities in the Colab notebook:

- Read a complex number from the keyboard.
- Find the complex conjugate of the complex number.
- Addition, subtraction, multiplication, and divisions of complex numbers.
- understands the concept of a complex number as a 2D vector.
- Find the length or magnitude of a complex number.

Section 5

Account Creation on IBM Q Experience Create your login id https://quantum-computing.ibm.com/ After Login, go through the interface with the following two options:

– Launch Composer: working with IBM composer (Drag and drop Options).

– Working with IBM Qiskit (refer to https://qiskit.org/textbook/content/ch-ex/)

SECOND YEAR

SEMESTER-III

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Elective I		CC/PCC	4	3	1	0	0
2	Elective II		CC/PCC	4	3	1	0	0
3	Technical Scientific Writing		AECC/HSM	2	0	0	0	4
4	Customer Relationship Management using Salesforce		AECC/HSM	0	1	0	2	0
5	Research Project I		PROJECT	8	0	0	0	16
			Total Credit:	18				

Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1		Bioinformatics
2		Information Theory Coding
3		Mobile Computing
Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1		Human Computer Interaction
2		Big Data Analytics
3		Industry 5.0

ELECTIVE I: BIOINFORMATICS

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Bioinformatics	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXXX	**Semester:** 3^{rd}

➠ Learning Objectives:

1. Adapt basic knowledge on various techniques and areas of applications in bioinformatics.
2. Analyze common problem in bioinformatics, alignment techniques, ethical issues, public data sources, and evolutionary modelling.
3. Discover the practical use of tools for specific bioinformatic areas

➠ Prerequisite:

A strong mathematical background- Proficiency with algorithm set theory, mathematical logic, Programming skills python, Perl, MATLAB, etc. and critical thinking and problem-solving skills.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction to bioinformatics	6	12.5%
Module-II: Pairwise sequence alignment	8	16%
Module-III: Multiple sequence alignment	8	17%
Module-IV: Scoring matrices	6	12.5%
Module-V: Database search methods	8	17%
Module-VI: Neural Networks and Hidden Markov Model	12	25%

➠ Syllabus Outline: .....

Module I: Introduction to bioinformatics: [6L]

Scope and applications of bioinformatics, Alignment of pairs of sequences; Introduction-Definition of sequence alignment, Methods - Dot matrix sequence comparison.

Module II: Pairwise sequence alignment: [8L]

Dynamic programming algorithm for sequence alignment – Global Alignment: Needleman-Wunsch, Local Alignment: Smith-Waterman , Gap penalty, Assessing the significance of an alignment.

Module III: Multiple sequence alignment: [8L]

Dynamic programming, progressive methods, Iterative methods, MSA using CLUSTAL W, PILEUP and CLUSTAL X, purpose and applications of multiple sequence alignment.

Module IV: Scoring matricese: [6L]

Similarity searches-PAM and BlOSUM matrix, Dayhoff mutation matrix, construction of PAM and BLOSUM matrix. Differences between PAM & BLOSUM.

Module V: Database search methods: [8L]

Database searching for similar sequences. Sequence similarity search, FASTA sequence database similarity search, BLAST sequence database similarity search, other methods of comparing database of sequences and patterns.

Module VI: Neural Networks and Hidden Markov Model: [12L]

The Theory -Introduction – Priors & likelihoods - Learning algorithms: backpropagation - Neural Networks: Applications - Sequence encoding & output interpretation- Sequence correlations & neural networks, The Theory - Introduction -Prior information & initialization -Likelihood & basic algorithms, Learning algorithms -Applications of HMMs: general aspects -Protein applications, The Theory -Introduction – Priors & likelihoods - Learning algorithms: backpropagation - Neural Networks: Applications - Sequence encoding & output interpretation- Sequence correlations & neural networks, The Theory - Introduction -Prior information & initialization -Likelihood & basic algorithms-Learning algorithms -Applications of HMMs: general aspects -Protein applications.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. Bioinformatics: Sequence and Genome Analysis David W. Mount, David Mount
2. Bioinformatics: the Machine Learning Approach – Pierre Baldi and Søren Brunak Publisher: MIT Press.

Reference Books:

1. Hooman H Rashidi, Lukas K Buehler. Bioinformatics Basics. -2000.
2. Per Jambeck, Cynthia Gibas. Developing Bioinformatics Computer Skills. Computers – 2001.
3. Bioinformatics Methods and Protocols: Methods and Protocols. edited by Stephen Misener, Stephen A Krawetz - Science – 1999.

Course Outcome (CO):

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

CO1: *Explain* the fundamental concepts of bioinformatics, sequence alignment basics, and dot-matrix analysis.

CO2: *Apply* dynamic programming algorithms (Needleman–Wunsch, Smith–Waterman) with appropriate scoring schemes and gap penalties for pairwise sequence alignment.

CO3: *Analyze* multiple sequence alignment approaches and tools (CLUSTALW, CLUSTALX, PILEUP) for biological sequence comparison.

CO4: *Evaluate* different scoring matrices (PAM, BLOSUM, Dayhoff) and determine their suitability for various similarity search tasks.

CO5: *Design* and *implement* sequence search and prediction models using FASTA/BLAST, Neural Networks, and Hidden Markov Models for biological data analysis.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	-	1	-
CO2	2	2	3	2	2	1
CO3	2	3	3	1	3	1
CO4	2	3	3	1	2	1
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ELECTIVE I: INFORMATION THEORY AND CODING

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Information Theory and Coding	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXXX	**Semester:** 3^{rd}

➠ Learning Objectives:

This course gives brief knowledge about the basic algebraic relationships of entropy, relative entropy, and mutual information. In this course students are going to learn how to compress the data using source coding and how to make data transmission reliable using channel coding. It introduces the basic principles of encoding, decoding, error detecting and error correcting techniques.

➠ Prerequisite: The students should have Computer Organization basics.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Basics of information theory	6	13%
Module-II: Channels	6	13%
Module-III: Error Correction	6	13%
Module-IV: Linear block codes	8	16%
Module-V: Cyclic Codes Polynomials	8	16%
Module-VI: BCH Codes and Tree CODES	14	29%

➠ Syllabus Outline: .....

Module I: Basics of information theory: [6L]

Basics of information theory, entropy for discrete ensembles; Shannon's noiseless coding theorem; Encoding of discrete sources.

Module II: Channels: [6L]

Channel models, channel capacity, channel coding, information capacity theorem, The Shannon limit.

Module III: Error Correction: [6L]

Linear and Block Codes for Error Correction

Module IV: Linear block codes: [8L]

Matrix description of linear block codes, equivalent codes, parity check matrix, decoding of a linear block code, perfect codes, Hamming codes.

Module V: Cyclic Codes Polynomials: [8L]

Cyclic Codes Polynomials, division algorithm for polynomials, a method for generating cyclic codes, matrix description of cyclic codes, Golay codes.

Module VI: BCH Codes and Tree CODES: [14L]

BCH Codes, Primitive elements, minimal polynomials, generator polynomials in terms of minimal polynomials, examples of BCH codes, Tree codes, trellis codes, polynomial description of convolutional codes, distance notions for convolutional codes, the generating function, matrix representation of convolutional codes, decoding of convolutional codes, distance and performance bounds for convolutional codes, examples of convolutional codes, Turbo codes, Turbo decoding.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. N. Abramson, Information and Coding, McGraw Hill, 1963.
2. M. Mansurpur, Introduction to Information Theory, McGraw Hill, 1987.

Reference Books:

1. R.B. Ash, Information Theory, Prentice Hall, 1970
2. Fundamentals of Information Theory and Coding Design, Roberto Togneri and Christopher J.S. deSilva, CRC Press, 2002.
3. Elements of Information Theory, Thomas M. Cover and Joy A. Thomas, 2nd ed., Wiley, 2006.

Course Outcome (CO):

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

CO1: *Explain* the fundamental concepts of entropy, information measures, Shannon's theorems, and discrete source encoding.

CO2: *Apply* channel models, channel capacity formulas, and information capacity theorems to evaluate communication system performance.

CO3: *Analyze* error-correcting codes including linear and block codes to determine their error-detection and correction capabilities.

CO4: *Evaluate* linear block codes, cyclic codes, parity-check methods, generator polynomials, and advanced algebraic coding structures.

CO5: *Design* and *implement* BCH, convolutional, trellis, and turbo coding/decoding schemes and assess their performance for reliable communication systems.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	-	-	-
CO2	2	2	3	2	1	-
CO3	2	2	3	2	2	1
CO4	2	3	3	2	2	2
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ELECTIVE I: MOBILE COMPUTING

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Mobile Computing	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXXX	**Semester:** 3^{rd}

➠ Learning Objectives:

1. To learn about various wireless & cellular communication networks and various telephone and satellite networks.
2. To build knowledge on various Adhoc and sensor networks routing protocol and energy efficient protocol.
3. To build skills in working with Cognitive radio networks and recent telecommunication networks
4. To design and development of various network protocol using simulation tools.

➠ Prerequisite:

Students should have knowledge of computer networks, including OSI/TCP-IP models, routing, and wireless communication basics. Basic understanding of probability, signal propagation, and electromagnetic concepts, Knowledge of operating systems, mobility concepts, and resource management. Programming experience and familiarity with network simulation tools (NS2/NS3, OMNET++, or MATLAB).

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction	10	20%
Module-II: Location and handoff management	10	21%
Module-III: Wireless transmission fundamentals	10	21%
Module-IV: Mobile Ad-hoc networks	6	12%
Module-V: Wireless sensor networks	6	13%
Module-VI: Cognitive radio networks	6	13%

Syllabus Outline:

Module I: Introduction: [10L]

Overview of wireless and mobile infrastructure; Preliminary concepts on cellular architecture; Design objectives and performance issues; Radio resource management and interface; Propagation and path loss models; Channel interference and frequency reuse; Cell splitting; Channel assignment strategies; Overview of generations:- 1G to 5G.

Module II: Location and handoff management: [10L]

Introduction to location management (HLR and VLR); Mobility models characterizing individual node movement (Random walk, Fluid flow, Markovian, Activity based); Mobility models characterizing the movement of groups of nodes (Reference point-based group mobility model, Community based group mobility model); Static (Always vs. Never update, Reporting Cells, Location Areas) and Dynamic location management schemes (Time, Movement, Distance, Profile Based); Terminal Paging (Simultaneous paging, Sequential paging); Location management and Mobile IP; Overview of handoff process; Factors affecting handoffs and performance evaluation metrics; Handoff strategies; Different types of handoffs (soft, hard, horizontal, vertical).

Module III: Wireless transmission fundamentals: [10L]

Introduction to narrow and wideband systems; Spread spectrum; Frequency hopping; Introduction to MIMO; MIMO Channel Capacity and diversity gain; Introduction to OFDM; MIMO-OFDM system; Multiple access control (FDMA, TDMA, CDMA, SDMA); Wireless local area network; Wireless personal area network (Bluetooth and zigbee).

Module IV: Mobile Ad-hoc networks: [6L]

Characteristics and applications; Coverage and connectivity problems; Routing in MANETs.

Module V: Wireless sensor networks: [6L]

Concepts, basic architecture, design objectives and applications; Sensing and communication range; Coverage and connectivity; Sensor placement; Data relaying and aggregation; Energy consumption; Clustering of sensors; Energy efficient Routing (LEACH).

Module VI: Cognitive radio networks: [6L]

Fixed and dynamic spectrum access; Direct and indirect spectrum sensing; Spectrum sharing; Interoperability and co-existence issues; Applications of cognitive radio networks. Introduction to D2D communications; High level requirements for 5G architecture; Introduction to the radio resource management, power control and mode selection problems; Millimeter wave communication in 5G.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Jochen Schiller, Mobile Communications, 2nd Edition, Pearson Education, 2020.
2. Andrea Goldsmith, Wireless Communications. Cambridge University Press, 2012.

Reference Books:

1. Ivan Stojmenovic, Handbook of Wireless Networking and Mobile Computing, Wiley, 2002.
2. Raj Kamal, Mobile Computing, 3rd Edition, Oxford University Press, 2019.
3. Asoke K. Talukder and Roopa R. Yavagal, Mobile Computing: Technology, Applications and Service Creation, 2nd Edition, McGraw-Hill, 2018.
4. William Stallings, Wireless Communications and Networks, 2nd Edition, Pearson Education, 2019.
5. K. N. Raja Rao, Advanced Mobile Wireless Communications: From Basic Concepts to 5G Technologies and IoT, Wiley, 2023.
6. Wei Xiang, Kan Zheng, and Xuemin (Sherman) Shen (Eds.), 5G Mobile and Wireless Communications Technology, Cambridge University Press, 2017.
7. Andreas F. Molisch, Wireless Communications, 3rd Edition, Wiley, 2023.
8. Biplab K. Sikdar, Mobile and Wireless Communications, PHI Learning, 2020.
9. Mischa Schwartz, Mobile Wireless Communications, Cambridge University Press, 2020.

Course Outcome (CO):

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

CO1: *Explain* the architecture, performance issues, and radio resource management principles of wireless and mobile communication systems.

CO2: *Apply* mobility models and location management techniques to evaluate handoff strategies in diverse mobile environments.

CO3: *Analyze* wireless transmission techniques, including MIMO, OFDM, and multiple-access mechanisms, for varied wireless network scenarios.

CO4: *Evaluate* routing protocols and energy-efficient mechanisms for Mobile Ad-hoc Networks and Wireless Sensor Networks.

CO5: *Design* advanced mobile communication solutions integrating cognitive radio techniques, spectrum

sensing, and 5G radio-resource management.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	1	2	2	1	-
CO2	2	-	2	2	2	-
CO3	3	1	3	3	2	-
CO4	3	-	3	3	3	2
CO5	3	2	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ELECTIVE II: HUMAN COMPUTER INTERACTION

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Human Computer Interaction	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXXX	**Semester:** 3^{rd}

Learning Objectives:

After completing this course, students will be able to:

1. Provide the basic knowledge on the levels of interaction, design models, techniques and validations focusing on the different aspects of human-computer interface and interactions.
2. Make the learners to think in design perspective and to evaluate interactive design.
3. Use the concepts and principles of HCI to analyze and propose solution for real life applications.
4. Become familiar with recent technology trends and challenges in HCI domain.

Prerequisite:

Students are expected to have basic knowledge of computer systems, operating systems, and software development, understanding of user interfaces, web/app development, and system design concepts, familiarity with human cognition fundamentals (memory, perception) is helpful, basic skills in problem-solving, modeling, and analytical reasoning, exposure to research methodology and usability testing is desirable but not mandatory.

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: HCI Foundations and Designing Interaction	10	21%
Module-II: Interaction Design Models	10	21%
Module-III: Guide Lines in HCI	6	12%
Module-IV: Collaboration and Communication	8	17%
Module-V: Human Factors and Security	6	12%
Module-VI: Validation and Advanced Concepts	8	17%

Syllabus Outline:

Module I: HCI Foundations and Designing Interaction: [10L]

Input–output channels, Human memory, Thinking: reasoning and problem solving, Emotion, Individual differences, Psychology and the design of interactive systems, Text entry devices, Positioning, pointing and drawing, Display devices, Devices for virtual reality and 3D interaction, Physical controls, sensors andspecial devices, Paper: printing and scanning. Overview of Interaction Design Models, Discovery - Framework, Collection - Observation, Elicitation, Interpretation - Task Analysis, Storyboarding, Use Cases, Primary Stakeholder Profiles, Project Management Document

Module II: Interaction Design Models: [10L]

Model Human Processor - Working Memory, Long-Term Memory, Processor Timing, Keyboard Level Model - Operators, Encoding Methods, Heuristics for M Operator Placement, What the Keyboard Level Model Does Not Model, Application of the Keyboard Level Model, GOMS - CMN-GOMS Analysis, Modeling Structure, State Transition Networks - Three-State Model, Glimpse Model, Physical Models, Fitts" Law

Module III: Guide Lines in HCI: [6L]

Shneideman's eight golden rules, Norman's Sever principles, Norman's model of interaction, Nielsen's ten heuristics, Heuristic evaluation, contextual evaluation, Cognitive walk-through

Module IV: Collaboration and Communication: [8L]

Face-to-face Communication, Conversation, Text-based Communication, Group working, Dialog design notations, Diagrammatic notations, Textual dialog notations, Dialog semantics, Dialog analysis and design.

Module V: Human Factors and Security: [6L]

Groupware, Meeting and decision support systems, Shared applications and artifacts, Frameworks for groupware Implementing synchronous groupware, Mixed, Augmented and Virtual Reality.

Module VI: Validation and Advanced Concepts: [8L]

Validations - Usability testing, Interface Testing, User Acceptance Testing Past and future of HCI: the past, present and future, perceptual interfaces, context-awareness and perception.

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. A Dix, Janet Finlay, G D Abowd, R Beale., Human-Computer Interaction, 3rd Edition, Pearson Publishers, 2008.

Reference Books:

1. Shneiderman, Plaisant, Cohen and Jacobs, Designing the User Interface: Strategies for Effective Human Computer Interaction, 5th Edition, Pearson Publishers, 2010.
2. Hans-Jorg Bullinger,“ Human-Computer Interaction”, Lawrence Erlbaum Associates, Publishers.
3. Jakob Nielsen,“ Advances in Human-computer Interaction”,Ablex Publishing Corporation.
4. Thomas S. Huang,“ Real-Time Vision for Human-Computer Interaction”, Springer.

Course Outcome (CO):

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

CO1: *Explain* the foundational concepts of human cognition, interaction models, and psychological principles underlying interactive system design.

CO2: *Apply* interaction design frameworks, task analysis methods, GOMS/KLM models, and Fitts' law to analyze and design effective user interactions.

CO3: *Evaluate* user interfaces using established HCI guidelines, heuristic principles, cognitive walkthroughs, and contextual evaluation techniques.

CO4: *Analyze* collaboration and communication models for designing multi-user, multimodal, and group-based interactive systems.

CO5: *Design* and *validate* advanced interactive systems incorporating usability testing, mixed/augmented/virtual reality concepts, and emerging HCI technologies.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	2	-	1	-
CO2	2	2	3	2	2	1
CO3	2	3	3	1	2	1
CO4	2	2	3	1	2	2
CO5	3	3	3	2	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ELECTIVE II: BIG DATA ANALYTICS

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Big Data Analytics	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXX	**Semester:** 3^{rd}

➠ Learning Objectives:

After completing this course, students will be able to:

1. Understand the need of Big Data, challenges and different analytical architectures
2. Installation and understanding of Hadoop Architecture and its ecosystems
3. Process Big Data with Advanced architectures like Spark.
4. Describe graphs and streaming data in Spark

➠ Prerequisite: Basic knowledge of computation.

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Introduction To Big Data	4	8%
Module-II: Hadoop Framework	8	16%
Module-III: Hadoop Ecosystem	8	17%
Module-IV: Spark Framework	8	17%
Module-V: Data Analysis with Spark Shell	10	21%
Module-VI: Spark SQL and GraphX	10	21%

➠ Syllabus Outline: .....

Module I: Introduction To Big Data : [4L]

Data Storage and Analysis - Characteristics of Big Data – Big Data Analytics - Typical Analytical Architecture – Requirement for new analytical architecture – Challenges in Big Data Analytics –Need of big data frameworks.

Module II: Hadoop Framework: [8L]

Hadoop – Requirement of Hadoop Framework - Design principle of Hadoop –Comparison with other system - Hadoop Components – Hadoop 1 vs Hadoop 2 – Hadoop Daemon's – HDFS Commands – Map Reduce Programming: I/O formats, Map side join, Reduce Side Join, Secondary sorting, Pipelining MapReduce jobs.

Module III: Hadoop Ecosystem: [8L]

Introduction to Hadoop ecosystem technologies: Serialization: AVRO, Co-ordination: Zookeeper, Databases: HBase, Hive, Scripting language: Pig, Streaming: Flink, Storm

Module IV: Spark Framework: [8L]

Introduction to GPU Computing, CUDA Programming Model, CUDA API, Simple Matrix, Multiplication in CUDA, CUDA Memory Model, Shared Memory Matrix Multiplication, Additional CUDA API Features.

Module V: Data Analysis with Spark Shell: [10L]

Writing Spark Application - Spark Programming in Scala, Python, R, Java - Application Execution.

Module VI: Spark SQL and GraphX: [10L]

SQL Context – Importing and Saving data – Data frames – using SQL – GraphX overview – Creating Graph – Graph Algorithms. Overview of Spark Streaming – Errors and Recovery – Streaming Source – Streaming live data with spark

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Mike Frampton, "Mastering Apache Spark", Packt Publishing, 2015.
2. Tom White,"Hadoop:The Definitive Guide",O'Reilly,4th Edition,2015.
3. Nick Pentreath, Machine Learning with Spark,Packt Publishing,2015.

Reference Books:

1. Mohammed Guller, Big Data Analytics with Spark, Apress,2015
2. Donald Miner, Adam Shook, "Map Reduce Design Pattern", O'Reilly, 2012

Course Outcome (CO):

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

CO1: *Explain* the fundamental characteristics and challenges of Big Data by differentiating it from traditional data storage and analysis.

CO2: *Implement* MapReduce programs and HDFS commands to solve a given data processing task.

CO3: *Compare* the roles of different Hadoop ecosystem technologies (e.g., Hive, HBase, Pig) to determine their suitability for various data analysis requirements.

CO4: *Defend* the selection of Spark over MapReduce based on performance metrics for in-memory processing.

CO5: *Design* a complete end-to-end Big Data solution that integrates Spark, Spark SQL, and Spark Streaming for a real-world scenario.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	1	1	2	1	2	1
CO2	2	1	3	2	2	1
CO3	2	2	3	2	3	2
CO4	3	3	3	3	3	2
CO5	3	3	3	3	3	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

ELECTIVE II: INDUSTRY 5.0

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P
Name: Industry 5.0	**Course Credit:** 04[3-1-0]
Department: Computer Science Engineering	**Category:** CC/PCC
Code: XXXXXXXX	**Semester:** 3^{rd}

Learning Objectives:

1. This course is aimed at giving basic understanding about the Digital marketing
2. This course is aimed at familiarizing the different styles & strategies of Digital Marketing
3. This course is aimed at providing plans and campaigns that are digitally becoming more prevalent in the current scenario.

Prerequisite: Fundamentals of Management

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Overview	6	12%
Module-II: Search Engine Optimization (SEO)	8	16%
Module-III: Social Media Marketing (SMM)	6	13%
Module-IV: Content Marketing	6	13%
Module-V: Online Advertising	10	21%
Module-VI: Email Marketing and Mobile Marketing	12	25%

Syllabus Outline: .....

Module I: Overview: [6L]

About Digital Marketing, Difference between Traditional Marketing and Digital Marketing, Benefits of using digital media, Inbound and Outbound Marketing, Online marketing POEM: (Paid, Owned, and Earned Media), Components of Online Marketing (Email, Forum, Social network, Banner, Blog).

Module II: Search Engine Optimization (SEO): [8L]

About SEO, Need of an SEO friendly website, Search Engine, Role of Keywords in SEO, Off-page Optimization, On-page Optimization concepts, Organic SEO vs Non-organic SEO.

Module III: Social Media Marketing (SMM): [6L]

About Social Media Marketing, Different types of Social Media Marketing

Module IV: Content Marketing: [6L]

About Content Marketing, Goals of Content Marketing, Types of Contents, etc.

Module V: Online Advertising: [10L]

About Online Advertising, Advantages of Online Advertising, Paid versus Organic, Pay Per Click (PPC) Model. Basic concepts CPC, PPC, CPM, CTR, CR

Module VI: Email Marketing and Mobile Marketing: [12L]

About Email marketing, Email newsletters, Digests, Dedicated Emails, Lead Nurturing, Sponsorship Emails and Transactional Emails, Drawbacks of Email Marketing, About Mobile Marketing, Objectives of Mobile Advertising, Creating a Mobile Marketing Strategy, About SMS Marketing.

➠ Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books:

Text Books:

1. Vandana Ahuja, Digital Marketing, 1st edition, Oxford

Reference Books:

1. Prof. Surabhi Singh, Digital Marketing, New Edition Mewar, University Press

➠ Course Outcome (CO):

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

CO1: *Explain* the fundamentals of digital marketing, online marketing models (POEM), and key differences between traditional and digital marketing.

CO2: *Apply* search engine optimization (SEO) techniques, including keyword analysis, on-page and off-

page optimization, and organic vs. inorganic ranking strategies.

CO3: *Analyze* the role of social media marketing (SMM) across platforms and evaluate their effectiveness for different business goals.

CO4: *Evaluate* content marketing strategies, types of digital content, and their alignment with business objectives.

CO5: *Design* and *assess* online advertising, PPC models, email marketing workflows, and mobile marketing strategies for real-world digital ecosystems.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	2	2	1	-	-	-
CO2	2	1	2	-	-	-
CO3	2	2	2	-	1	-
CO4	2	2	2	-	1	-
CO5	3	2	3	1	2	1

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

TECHNICAL SCIENTIFIC WRITING

Course Information:

School: School of Science & Technology	**Course Type:** L-T-P-S
Name: Technical Scientific Writing	**Course Credit:** 02[0-0-0-4]
Department: Computer Science Engineering	**Category:** AECC/HSM
Code: XXXXXXXX	**Semester:** 3^{rd}

Learning Objectives:

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

1. Understand that how to improve your writing skills and level of readability
2. Learn about what to write in each section
3. Understand the skills needed when writing a Title Ensure the good quality of paper at very first-time submission

Prerequisite: Fundamentals of English Grammar

Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: Planning and Preparation	4	16
Module-II: Structure of the Paper	4	16
Module-III: Key Skills	4	17
Module-IV: Results and Discussion	4	17
Module-V: Submission	4	17
Module-VI: Guest Lecture from R& D organizations	4	17

Syllabus Outline:

Module I: Planning and Preparation: [4L]

Word Order, Breaking up long sentences, Structuring Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness.

Module II: Structure of the Paper: [4L]

Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticising, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts. Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check.

Module III: Key Skills: [4L]

Key skills are needed when writing a Title; key skills are needed when writing an Abstract, key skills are needed when writing an Introduction, skills needed when writing a Review of the Literature.

Module IV: Results and Discussion: [4L]

Skills are needed when writing the Methods, skills needed when writing the Results, skills are needed when writing the Discussion, skills are needed when writing the Conclusions.

Module V: Submission: [4L]

Useful phrases, how to ensure paper is as good as it could possibly be the first- time submission.

Module VI: Guest Lecture from R& D organizations: [4L]

Contemporary Issues.

➠ Teaching–Learning Methodology: .....

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)
- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

➠ Text & Reference Books: .....

Text Books:

1. Goldbort R (2006) Writing for Science, Yale University Press.

Reference Books:

1. Day R (2006) How to Write and Publish a Scientific Paper, Cambridge University Press.
2. "Technical Communication" by Mike Markel and Stuart A. Selber
3. "Technical Writing for Engineers & Scientists" by Leo Finkelstein
4. "Scientific Writing and Communication: Papers, Proposals, and Presentations" by Angelika H. Hofmann
5. "The Elements of Style" by William Strunk Jr. and E.B. White
6. "The Craft of Scientific Writing" by Michael Alley
7. "The Chicago Manual of Style" by The University of Chicago Press Editorial Staff

Course Outcome (CO):

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

CO1: *Plan* and *structure* clear, concise, and unambiguous scientific text.

CO2: *Write* well-organized research papers following standard scientific formats.

CO3: *Apply* ethical research writing practices including paraphrasing and plagiarism avoidance.

CO4: *Critically* present and discuss research results using appropriate academic language.

CO5: *Prepare* and submit high-quality research manuscripts and technical reports.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	-	2	-	-	-	2
CO2	-	3	-	-	-	3
CO3	-	2	-	-	-	3
CO4	-	2	-	-	-	2
CO5	-	3	-	-	-	3

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

CUSTOMER RELATIONSHIP MANAGEMENT USING SALESFORCE

➠ Course Information:

School: School of Science & Technology	**Course Type:** L-T-P-S
Name: Customer Relationship Management using Salesforce	**Course Credit:** 00[1-0-2-0]
Department: Computer Science Engineering	**Category:** AECC/HSM
Code: XXXXXXXX	**Semester:** 3^{rd}

➠ Learning Objectives:

1. To understand the concepts and importance of Customer Relationship Management (CRM).
2. To familiarize students with Salesforce as a leading cloud-based CRM solution.
3. To provide practical experience in customizing and extending CRM functionality.
4. To enable the development and integration of cloud-based business applications.
5. To introduce cloud architecture and the Salesforce platform for scalable enterprise solutions.

➠ Prerequisite:

Students should have basic knowledge of Web Technologies, Databases, and Software Engineering

➠ Course Content/ Syllabus Table:

Module No.	No. of Lecture / Contact hour	Weightage (%)
Module-I: CRM and Cloud Computing Basics	4	11%
Module-II: Salesforce Platform Overview	4	11%
Module-III: Declarative Development and Automation	6	17%
Module-IV: Programmatic Development with Apex	8	22%
Module-V: Lightning Web Components and App Development	6	17%
Module-VI: Integration and Analytics	8	22%

➠ Syllabus Outline: .....

Module I: CRM and Cloud Computing Basics : [4L]

- Introduction to CRM: Evolution, types, and business relevance
- CRM components: Operational, Analytical, Collaborative
- Introduction to Cloud Computing: IaaS, PaaS, SaaS
- CRM as SaaS: Overview of market leaders

Module II: : Salesforce Platform Overview : [4L]

- Salesforce architecture and editions
- Navigating Salesforce Lightning interface
- Standard objects and their relationships (Account, Contacts, Leads)
- Custom objects and schema builder

Module III: Declarative Development and Automation: [6L]

- Customizing page layouts and record types
- Validation rules, workflow rules, and process builder
- Flows and approval processes
- Salesforce App Builder and Lightning App setup

Module IV: Programmatic Development with Apex: [8L]

- Apex syntax and data types
- SOQL and SOSL queries
- Writing triggers and classes
- Introduction to asynchronous Apex (future, batch, queueable)

Module V: Lightning Web Components and App Development: [6L]

- Component-driven development with LWC
- Events and data binding
- Creating reusable components
- Deployment using change sets and unmanaged packages

Module VI: Integration and Analytics: [8L]

- REST API and external integrations
- AppExchange and prebuilt integrations
- Reports, dashboards, and Einstein Analytics
- Final project: CRM application for a fictional business

Teaching–Learning Methodology:

- ***Pedagogy for Course Delivery:*** Hybrid Mode (Offline Class/ Presentation/ Video/ MOOC)

- ***Continuous Assessment:*** Quiz/ Assessment/ Presentation/ Problem solving etc.

Text & Reference Books:

Text Books:

1. Paul Battisson, Learning Salesforce Development with Apex, Packt Publishing
2. Salesforce Official Documentation: https://developer.salesforce.com/docs

Reference Books:

1. Siddhesh Kabe, Salesforce Essentials for Administrators, Packt Publishing
2. Michael Wicherski, Advanced Apex Programming, Salesforce Press
3. Trailhead (Salesforce's Official Learning Platform): https://trailhead.salesforce.com

Course Outcome (CO):

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

CO1: *Explain* the fundamentals, architecture, and evolution of Customer Relationship Management (CRM) systems with emphasis on the Salesforce ecosystem.

CO2: *Apply* Salesforce CRM features to manage sales, service, and marketing workflows in real-world business scenarios.

CO3: *Design* and *implement* process automation using validation rules, workflow rules, and Flow Builder to streamline business operations.

CO4: *Analyze* Salesforce reports and dashboards to extract actionable business insights and support decision-making.

CO5: *Able* to develop and *demonstrate* a mini-project simulating a real-world CRM scenario and prepare for Salesforce Administrator (ADM 201) certification.

CO-PO-PSO Mapping:

Course Outcomes	Programme Outcomes			Programme Specific Outcomes		
	PO1	PO2	PO3	PSO1	PSO2	PSO3
CO1	-	-	2	2	-	-
CO2	2	-	2	3	-	-
CO3	3	-	2	3	-	-
CO4	2	-	2	2	2	-
CO5	3	2	3	3	2	2

1: Slight (Low) *2: Moderate (Medium)* *3: Substantial (High)*

SEMESTER-IV

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Dissertation/ Research Project II		PROJECT	16	0	0	0	32
2	Grand Viva		SEC/PSE	2	2	0	0	0
			Total Credit:	18