Sister Nivedita University

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

School of Engineering
Department of Computer Science & Engineering

SYLLABUS
FOR
MASTER OF TECHNOLOGY (M. TECH) IN COMPUTER SCIENCE AND ENGINEERING

Regulations (R24) [NEP]

R24–25 Academic Session

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

Year	Semester	hrs./Week	Credit
1st	1st	32	28
	2nd	23	22
2nd	3rd	25	18
	4th	32	18
Total			**86**

Category Definition

Definition of Category/Type	Abbreviation
Professional Core	**PCC**
Professional Core Elective	**PCE**
Discipline Specific Electives	**DSE**
Ability Enhancement Courses	**AEC**
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	4	3	1	0	0
2	Advanced Operating System		PCC	4	3	1	0	0
3	Advanced Data Structures		PCC	4	3	1	0	0
4	Object Oriented Analysis and Design		PCC	4	3	1	0	0
5	Distributed Systems/ Data Preparation and Analysis		PCE	4	3	1	0	0
6	Foreign Language I		AEC	2	2	0	0	0
7	Object Oriented Analysis and Design Lab		PCC	2	0	0	4	0
9	Advanced Data Structures Lab		PCC	2	0	0	4	0
Total Credit				**28 Credit**				

SEMESTER-II

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

SECOND YEAR

SEMESTER-III

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Elective I		PCC	4	3	1	0	0
2	Elective II		PCC	4	3	1	0	0
3	Technical Scientific Writing		AEC	2	2	0	0	0
4	Research Project 1		Project	8	0	0	16	0
Total Credit				**18 Credit**				

Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1	PGCSE301A	Bioinformatics
2	PGCSE301B	Information Theory Coding
3	PGCSE301C	Mobile Computing
Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1	PGCSE302A	Human Computer Interaction
2	PGCSE302B	Big Data Analytics
3	PGCSE302	Industry 5.0

SEMESTER-IV

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Dissertation/ Research Project		CC	16	0	0	0	32
2	Grand Viva		AEC	2	2	0	0	0
Total Credit				**18 Credit**				

FIRST YEAR

SEMESTER-I

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Mathematical Foundations of Computer Science		DSE	4	3	1	0	0
2	Advanced Operating System		PCC	4	3	1	0	0
3	Advanced Data Structures		PCC	4	3	1	0	0
4	Object Oriented Analysis and Design		PCC	4	3	1	0	0
5	Distributed Systems/ Data Preparation and Analysis		PCE	4	3	1	0	0
6	Foreign Language I		AEC	2	2	0	0	0
7	Object Oriented Analysis and Design Lab		PCC	2	0	0	4	0
9	Advanced Data Structures Lab		PCC	2	0	0	4	0
Total Credit=28								

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Mathematical Foundations of Computer Science	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** DSE
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2										1
CO2	3	2										
CO3	2	1										
CO4	2	1										1
CO5	2	1										
CO6	3	2										1
Avg.	3	2										1

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

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.

CO6: Use Max-flow Min-cut theorem, Ford and Fulkerson algorithm to design complex engineering problems.

ADVANCED OPERATING SYSTEM

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Advanced Operating System	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

1) To study, 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: Basics of programming language, Logic building skills.

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	10%
Module-II: Processes and processors in distributed systems	8	20%
Module-III: Distributed File Systems	8	20%
Module-IV: Distributed Shared Memory	8	20%
Module-V: Distributed Web-based Systems	8	10%
Module-VI: Distributed Deadlock Detection	8	20%

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 distributedsystems: 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: [8L]

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: [8L]

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

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1									
CO2	3	3	3	1								1
CO3	3	3	3	2								1
CO4	3	3	3	1								1
CO5	3	3	3	2								1
CO6	3	3	3	2								1
Avg.	3	2.83	2.5	1.33								0.83

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: List the principles of distributed systems and describe the problems and challenges associated with these principles.

CO2: Understand Distributed Computing techniques, Synchronous and Processes.

CO3: Apply Shared Data access and Files concepts.

CO4: Understand Distributed File Systems and Distributed Shared Memory.

CO5: Apply Distributed web-based system.

CO6: Understand the concept of Distributed Deadlock Detection.

ADVANCED DATA STRUCTURES

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Advanced Data Structures	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

1) To impart the basic concepts of data structures and algorithms.
2) To understand concepts about searching and sorting techniques.
3) To understand basic concepts about stacks, queues, lists, trees and graphs.
4) To understanding about writing algorithms and step by step approach in
5) solving problems with the help of fundamental data structure

Prerequisite:

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction to Data Structures	6	10%
Module-II: Structures and Arrays	6	20%
Module-III: Linear Data Structure	12	20%
Module-IV: Non-Linear Data Structure	12	20%
Module-V: Sorting Algorithms	6	10%
Module-VI: Searching and Hashing	6	20%

SYLLABUS OUTLINE:

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

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1									
CO2	3	3	3	1								1
CO3	3	3	3	2								1
CO4	3	3	3	1								1
CO5	3	3	3	2								1
CO6	3	3	3	2								1
Avg.	3	2.83	2.5	1.33								0.83

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: To be able to classify linear and non-linear data structure.

CO2: To be able to solve different problems using Arrays.

CO3: To be able to make use of linked list for various operations on polynomials, sparse matrix etc.

CO4: To be able to utilize the knowledge of Stack, Queues in solving real life problem.

CO5: To be able to apply the knowledge of several binary trees in problem solving.

CO6: To be able to identify of the most appropriate searching or sorting algorithm for enhancing the efficiency (i.e. reduce the run-time) or for better memory utilization.

OBJECT ORIENTED ANALYSIS AND DESIGN

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Object Oriented Analysis and Design	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

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:

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand and apply the principles of Object-Oriented Analysis and Design, including the Unified Process and various UML diagrams.

CO2: Analyze and implement GRASP patterns and GoF design patterns to create efficient and maintainable object-oriented software designs.

CO3: Develop comprehensive use case models and domain models for real-world systems, demonstrating the ability to identify and describe conceptual classes, associations, and hierarchies.

CO4: Create and refine logical architectures using UML package diagrams, class diagrams, and interaction diagrams, applying appropriate design patterns to solve complex problems.

CO5: Translate object-oriented designs into code and develop effective testing strategies for object-oriented systems, including class testing, integration testing, GUI testing, and system testing.

CO6: Design and model the physical aspects of a system using UML component diagrams and deployment diagrams, considering processors, connections, components, and other system elements.

DISTRIBUTED SYSTEMS

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Distributed Systems	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

1) To understand the foundations of distributed systems.
2) To learn issues related to clock Synchronization and the need for global state in distributed systems.
3) To learn distributed mutual exclusion and deadlock detection algorithms.
4) To understand the significance of agreement, fault tolerance and recovery protocols in Distributed Systems.
5) To learn the characteristics of peer-to-peer and distributed shared memory systems

Prerequisite:

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	10%
Module-II: Design issues and challenges	8	20%
Module-III: Message Ordering & Snapshots	8	20%
Module-IV: Distributed Mutex & Deadlock	8	20%
Module-V: Recovery & Consensus	8	10%
Module-VI: P2P & Distributed Shared Memory	8	20%

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understanding Distributed System Models.

CO2: Analysing Distributed Computations.

CO3: Message Ordering and Global Snapshots.

CO4: Implementing Distributed Mutual Exclusion and Deadlock Detection.

CO5: Recovery and Consensus in Distributed Systems.

CO6: Exploring Peer-to-Peer and Distributed Shared Memory Systems.

DATA PREPARATION AND ANALYSIS

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Data Preparation and Analysis	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

1) The course introduces the methods for data preparation and data understanding.
2) It covers essential exploratory techniques for understanding multivariate data by summarizing it through statistical and graphical methods.
3) Supports to summarize use of predictive analytics, data science and datavisualization.

Prerequisite:

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction to Exploratory Data Analysis	8	10%
Module-II: Data Transformation	8	20%
Module-III: Correlation Analysis and Time Series	8	20%
Module-IV: Model Development, Data Summarization and Visualization	8	20%
Module-V: Clustering Algorithms	8	10%
Module-VI: Dimensionality Reduction	8	20%

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Handle missing data in the real-world data sets by choosing appropriate methods.

CO2: Summarize the data using basic statistics. Visualize the data using basic graphs and plots.

CO3: Understand the concept of Correlation Analysis and Time Series 4 hours Analysis.

CO4: Identify the outliers if any in the data set.

CO5: Choose appropriate feature selection and dimensionality reduction.

CO6: Apply Techniques for handling multi-dimensional data.

RESEARCH METHODOLOGY AND IPR

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Research Methodology and IPR	**COURSE CREDIT:** 02[2-0-0]
DEPARTMENT: Computer Science	**CATEGORY:** SEC
CODE: XXXXXX	**SEMESTER:** 1^{st}

THEORY

Learning objectives:

1) To demonstrate the identification of the research problems.
2) To make the awareness on the literature studies, plagiarism and ethics.
3) To train the knowledge on technical writing.
4) To analyze the nature of intellectual property rights and new developments
5) To facilitate the need of the patent rights.

Prerequisite:

Course content/ Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction	8	10%
Module-II: Literature studies and Plagiarism	8	20%
Module-III: Nature of Intellectual Property	8	20%
Module-IV: International Scenario	8	20%
Module-V: Patent Rights	8	10%
Module-VI: New Developments in IPR	8	20%

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.

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand research problem formulation.

CO2: Analyse research related information.

CO3: Demonstrate research ethics.

CO4: Explain the today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.

CO5: Discuss that when IPR would take such important place in growth of individuals & nation, it is needless to emphasise the need of information about Intellectual Property Right to be promoted among students in general & engineering.

CO6: Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.

OBJECT ORIENTED ANALYSIS AND DESIGN LAB

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P-S
NAME: Object Oriented Analysis and Design Lab	**COURSE CREDIT:** 02[0-0-4-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 1^{st}

List of Practical:

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 Engineering	**COURSE TYPE:** L-T-P-S
NAME: Advanced Data Structures Lab	**COURSE CREDIT:** 02[0-0-4-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 1^{st}

List of Practical:

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 Science		PCC	4	3	1	0	0
2	Advanced Software Engineering/ Web Analytics and Development		PCE	4	3	1	0	0
3	Fog and Edge Computing/ Soft Computing		PCE	4	3	1	0	0
4	Machine Learning/ Quantum Computing		PCE	4	3	1	0	0
5	Foreign Language II		AEC	2	2	0	0	0
6	Mini Project with Seminar / Industry Research Project		Project	3	0	0	4	0
7	Machine Learning Lab/ Quantum Computing Lab		PCE	1	0	0	2	0
Total Credit=22								

HIGH PERFORMANCE COMPUTING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: High Performance Computing	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

1) To provide knowledge on high performance computing concepts to the students.
2) To comprehend the students how to analyze the parallel programming through OpenMP, MPI, CUDA.
3) To teach the student how to apply job management techniques and evaluate the performance.

Prerequisite:

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction to High Performance Computing (HPC)	8	10%
Module-II: HPC Paradigms	8	20%
Module-III: Parallel Programming	8	20%
Module-IV: Job Management Systems	8	20%
Module-V: Achieving Performance	8	10%
Module-VI: HPC Benchmarks	8	20%

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

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: To knowledge the overview and analyse the performance metrics of high performance computing.

CO2: To comprehend the various High Performance Computing Paradigms and Job Management Systems.

CO3: To design and develop various applications with OpenMP, MPI and CUDA.

CO4: To analyse the benchmarks of high-performance computing.

CO5: To demonstrate the various emerging trends of high-performance computing.

CO6: To apply high performance computing concepts in problem solving.

ADVANCED SOFTWARE ENGINEERING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Advanced Software Engineering	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

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:

Course content/ Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Overview of Software Engineering and Software	8	10%
Module-II: Project Management	8	20%
Module-III: Modelling Requirements	8	20%
Module-IV: Software Design	8	20%
Module-V: Validation and Verification	8	10%
Module-VI: Software Evolution	8	20%

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]

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Apply the principles of the engineering processes in software development.

CO2: Demonstrate software project management activities such as planning, scheduling and Estimation.

CO3: Model the requirements for the software projects.

CO4: Design and Test the requirements of the software projects.

CO5: Implement the software development processes activities from requirements to validation and verification.

CO6: Apply and evaluate the standards in process and in product.

WEB ANALYTICS AND DEVELOPMENT

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Web Analytics and Development	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

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:

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.

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand Web Analytics Concepts-Grasp fundamental terms, processes, and the importance of web analytics.

CO2: Effectively gather and interpret data using web logs, beacons, and JavaScript tags.

CO3: Perform heuristic evaluations, site visits, and surveys for qualitative insights.

CO4: Utilize metrics such as bounce rate, page views, and unique visitors to assess website performance.

CO5: Enhance website and campaign performance through data-driven optimization techniques.

CO6: Implement and leverage Web Analytics 2.0 and Google Analytics for comprehensive data analysis.

FOG AND EDGE COMPUTING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Fog and Edge Computing	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

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		%
Module-II: Challenges in Federating Edge Resources		%
Module-III: Management and Orchestration of Network Slices in 5G, Fog, Edge, and Clouds		%
Module-IV: Optimization Problems in Fog and Edge Computing		%
Module-V: Middleware for Fog and Edge Computing and Technologies in Fog Computing		%
Module-VI: Applications and Issues		%

SYLLABUS OUTLINE:

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

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: [L]

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: [L]

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: [L]

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: [L]

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: [L]

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.

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												
CO6												
Avg.												

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand the principles, architectures of fog computing

CO2: Understand the communication and management of fogs

CO3: Understand storage and computation in fogs

CO4: Design and Implement Internet of Everything (IoE) applications through fog computing architecture.

CO5: Analysis the performance of the applications developed using fog architecture

CO6: Understand the security and privacy issues of fog computing

SOFT COMPUTING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Soft Computing	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

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: Mcpitt , 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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2								
CO2	3	3	3	3	2	1						3
CO3	2	3	3	3	2	2						2
CO4	1	2	3	3	3	1						2
CO5	1	3	3	3	2	2						3
CO6	1	2	3	3	3	2						3
Avg.	1.83	2.67	2.83	2.83	2.40	1.60						2.60

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: To Understand intelligent systems leveraging the paradigm of soft computing techniques.

CO2: To get the knowledge solutions by various soft computing approaches for finding the optimal solutions.

CO3: To Recognize the feasibility of applying a soft computing methodology for a particular problem.

CO4: To Design the methodology to solve optimization problems using fuzzy logic, genetic algorithms and neural networks.

CO5: To Design hybrid system to revise the principles of soft computing in various applications.

CO6: To Analyse the applications of Soft Computing Systems.

MACHINE LEARNING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Machine Learning	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

1) To understand the basic theory underlying machine learning.
2) To be able to formulate machine learning problems corresponding to different applications.
3) To understand a range of machine learning algorithms along with their strengths and weaknesses.
4) To be able to apply machine learning algorithms to solve problems of moderate complexity.
5) To 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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	1								1
CO2	3	2		2	1							2
CO3	2	2	3	2	2							2
CO4	1	2	3	3	2							3
CO5	2	2	3	3	3							3
CO6		2	3	3	2							
Avg.	2.20	2.17	2.80	2.33	2.00							2.20

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: To be able to discuss the basics of learning problems with hypothesis.

CO2: To be able to understand the features of machine learning to deal with real world problems.

CO3: To be able to differentiate the machine learning algorithms as supervised learning and unsupervised learning.

CO4: To be able to design and analyze various classification and clustering algorithms

CO5: To be able to develop and tune the machine learning models with datasets

CO6: To be able to evaluate the models for optimization engineering problems

QUANTUM COMPUTING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Quantum Computing	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

THEORY

Learning objectives:

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: Basic knowledge of Microprocessor and Micro controller.

Course content/Syllabus table:

Module No.	No. of lecture / Contact hour	Weightage (%)
Module-I: Introduction to Quantum Information		%
Module-II: Quantum Algorithms Basic		%
Module-III: Quantum Algorithms Advanced		%
Module-IV: Quantum True Random Number Generators (QTRNG)		%
Module-V: Basic Quantum key distribution		%
Module-VI: Advanced Quantum key distribution		%

SYLLABUS OUTLINE:

Module I: Introduction to Quantum Information: [L]

States, Operators, Measurements, Quantum Entanglement: Quantum Teleportation, Super-dense coding, CHSH Game, Quantum gates and circuits.

Module II: Quantum Algorithms Basic: [L]

Deutsch-Jozsa, Simon, Grover, Shor, Implication of Grover's and Simon's algorithms towards classical symmetric key cryptosystems.

Module III: Quantum Algorithms Advanced: [L]

Implication of Shor's algorithm towards factorization and Discrete Logarithm based classical public key cryptosystems.

Module IV: Quantum True Random Number Generators (QTRNG): [L]

Quantum True Random Number Generators (QTRNG): Detailed design and issues of quantumness, Commercial products and applications.

Module V: Basic Quantum key distribution: [L]

Quantum key distribution (QKD): BB84, Ekert, Semi-Quantum QKD protocols.

Module VI: Advanced Quantum key distribution: [L]

Variations in Semi-Quantum QKD protocols, Issuesof Device Independence, Commercial products.

- ***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-quantumcryptography

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	1	1						
CO2	3	3	3	2	1	1						1
CO3	3	3	3	2	1	1						
CO4	3	3	3	2	1	1						
CO5	3	3	3	2	1	1						
CO6	3	3	3	2	2							
Avg.	3	3	3	2.16	1.16	0.8						0.16

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand the basic concepts on quantum computing

CO2: Able to implement quantum algorithms for performing computations on quantum computers

CO3: Generate perfectly unpredictable random numbers to ensure the strongest level of encryption

CO4: Ensure secure communication using quantum key distribution method

CO5: Evaluate and standardize quantum-resistant public-key cryptographic algorithms

CO6: Perform quantum computations to solve simple problems

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	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

List of Practical

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 Engineering	**COURSE TYPE:** L-T-P
NAME: Quantum Computing Lab	**COURSE CREDIT:** 01[0-0-2]
DEPARTMENT: Computer Science	**CATEGORY:** PCE
CODE: XXXXXX	**SEMESTER:** 2^{nd}

List of Practical

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		PCC	4	3	1	0	0
2	Elective II		PCC	4	3	1	0	0
3	Technical Scientific Writing		AEC	2	2	0	0	0
4	Research Project 1		Project	8	0	0	16	0
				Total Credit=18				

Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1	PGCSE301A	Bioinformatics
2	PGCSE301B	Information Theory Coding
3	PGCSE301C	Mobile Computing
Elective – I		
Sl No.	**Paper Code**	**Paper Name**
1	PGCSE302A	Human Computer Interaction
2	PGCSE302B	Big Data Analytics
3	PGCSE302	Industry 5.0

BIOINFORMATICS

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Bioinformatics	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE301A	**SEMESTER:** 3^{rd}

THEORY

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		%
Module-II: Pairwise sequence alignment		%
Module-III: Multiple sequence alignment		%
Module-IV: Scoring matrices		%
Module-V: Database search methods		%
Module-VI: Neural Networks and Hidden Markov Model		%

SYLLABUS OUTLINE:

Module I: Introduction to bioinformatics: [L]

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: [L]

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: [L]

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: [L]

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

Module V: Database search methods: [L]

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: [L]

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	-	-	-	-	-	-	-	-	-
CO2	3	3	3	3	2	1	-	-	-	-	-	3
CO3	2	3	3	3	2	2	2	-	-	-	-	2
CO4	1	2	3	3	1	1	-	-	-	-	-	2
CO5	1	3	3	3	2	-	-	-	-	-	-	3
CO6	1	2	3	3	3	2	2	-	-	-	-	3
Avg	1.83	2.67	2.83	2.83	2.40	1.60	-	-	-	-	-	2.60

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Apply knowledge of bioinformatics in a practical project.

CO2: Develop the ability for critical assessment of scientific research publications in bioinformatics.

CO3: Build an understanding of the research process in general, such as research methods, scientific writing, and research ethics.

CO4: Evaluate the main databases at the NCBI and EBI resources

CO5: Compare the databases, tools, repositories and be able to use each one to extract specific information.

CO6: Demonstrate the selected tools at NCBI and EBI to run simple analyses on genomic sequences.

INFORMATION THEORY AND CODING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Information Theory and Coding	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE301B	**SEMESTER:** 3^{rd}

THEORY

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		%
Module-II: Channels		%
Module-III: Error Correction		%
Module-IV: Linear block codes		%
Module-V: Cyclic Codes Polynomials		%
Module-VI: BCH Codes and Tree CODES		%

SYLLABUS OUTLINE:

Module I: Basics of information theory: [L]

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

Module II: Channels: [L]

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

Module III: Error Correction: [L]

Linear and Block Codes for Error Correction

Module IV: Linear block codes: [L]

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: [L]

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: [L]

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.

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	-	-	-	-	-	-	-
CO2	3	3	3	1	1	-	-	-	-	-	-	1
CO3	3	3	3	1	-	-	-	-	-	-	-	1
CO4	-	-	-	-	-	-	-	-	-	-	-	1
CO5	-	-	-	-	-	-	-	-	-	-	-	1
CO6	-	-	-	-	-	-	-	-	-	-	-	1
Avg	1.5	1.33	1.16	0.5	0.3	-	-	-	-	-	-	0.8

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand entropy by measure of information content of the message

CO2: Apply entropy and oder of information sources to analyze Markov model

CO3: Understand uniqueness of code using Kraft Inequality and prefix code

CO4: Understand the conversion of information into binary sequence using Shanon, Shanon Fano and Huffman encoding algorithms

CO5: Model continuous and discrete communication channels using input, out and joint probability matrix

CO6: Determine channel capacity of binary symmetric and binary erasure channels using mutual information and Muroga's theorem

MOBILE COMPUTING

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Mobile Computing	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE301C	**SEMESTER:** 3^{rd}

THEORY

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: Basic knowledge of Networking

Course content/Syllabus table:

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

SYLLABUS OUTLINE:

Module I: Introduction: [L]

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: [L]

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: [L]

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: [L]

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

Module V: Wireless sensor networks: [L]

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: [L]

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.

- ***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. Pearson Education, 2009.
2) Andrea Goldsmith, Wireless Communications. Cambridge University Press, 2012.

Reference Books:

1) Ivan Stojmenovic, Handbook of Wireless Networking and Mobile Computing, Wiley, 2002.
2) Ezio Biglieri, Andrea J. Goldsmith, Larry J. Greenstein, Narayan Mandayam and H. Vincent Poor, Principles of Cognitive Radio. Cambridge University Press, 2012

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3								
CO2	3	3	3									1
CO3	3	3	3									
CO4	3	3	3									
CO5	3	3	3									
CO6	3	3	3									
Avg.	3	3	3	0.5								0.16

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Understand the working principles of mobile networks and Contrast different types of telecommunication networks.

CO2: Study on location, handoff management and wireless fundamentals.

CO3: Study on MANET and Sensor networks including architecture, routing and power optimization technique.

CO4: Study on cognitive ratio networks and its applications.

CO5: Assess the recent telecommunication networks, resource management

CO6: Design & development of various wireless network protocols using simulation tools

HUMAN COMPUTER INTERACTION

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Human Computer Interaction	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE302A	**SEMESTER:** 3^{rd}

THEORY

Learning objectives:

1) To 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) To make the learners to think in design perspective and to evaluate interactive design.
3) To use the concepts and principles of HCI to analyze and propose solution for real life applications.
4) To become familiar with recent technology trends and challenges in HCI domain.

Prerequisite: Basic knowledge of computation.

Course content/Syllabus table:

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

SYLLABUS OUTLINE:

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

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: [L]

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: [L]

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: [L]

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: [L]

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: [L]

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

- ***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.

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	-	-	-	-	-	-	-	-
CO2	3	3	3	3	2	1	-	-	-	-	-	3
CO3	2	3	3	3	2	2	-	-	-	-	-	2
CO4	1	2	3	3	1	-	-	-	-	-	-	2
CO5	1	3	3	3	2	-	-	-	-	-	-	3
CO6	1	2	3	3	3	2	-	-	-	-	-	3
Avg	1.83	2.67	2.83	2.83	2.40	1.60	-	-	-	-	-	2.60

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Enumerate the basic concepts of human, computer interactions and create the processes of human computer interaction life cycle.

CO2: Analyze and design the various interaction design models.

CO3: Apply the interface design standards/guidelines for evaluating the developed interactions.

CO4: Establish the different levels of communication across the application stakeholders.

CO5: Apply product usability evaluations and testing methods.

CO6: Demonstrate the principles of human computer interactions through the prototype modelling.

BIG DATA ANALYTICS

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Big Data Analytics	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE302B	**SEMESTER:** 3^{rd}

THEORY

Learning objectives:

1) To understand the need of Big Data, challenges and different analytical architectures
2) Installation and understanding of Hadoop Architecture and its ecosystems
3) Processing of 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		%
Module-II: Hadoop Framework		%
Module-III: Hadoop Ecosystem		%
Module-IV: Spark Framework		%
Module-V: Data Analysis with Spark Shell		%
Module-VI: Spark SQL and GraphX		%

SYLLABUS OUTLINE:

Module I: Introduction To Big Data : [L]

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: [L]

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: [L]

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

Module IV: Spark Framework: [L]

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: [L]

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

Module VI: Spark SQL and GraphX: [L]

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

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	-	-	-	-	-	-	-	-
CO2	3	3	3	3	2	1	-	-	-	-	-	3
CO3	2	3	3	3	2	2	-	-	-	-	-	2
CO4	1	2	3	3	1	-	-	-	-	-	-	2
CO5	1	3	3	3	2	-	-	-	-	-	-	3
CO6	1	2	3	3	3	2	-	-	-	-	-	3
Avg	1.83	2.67	2.83	2.83	2.40	1.60	-	-	-	-	-	2.60

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Discuss the challenges and their solutions in Big Data

CO2: Understand and work on Hadoop Framework and eco systems.

CO3: Explain and Analyse the Big Data using Map-reduce programming in Both Hadoop and Spark framework.

CO4: Demonstrate spark programming with different programming languages.

CO5: Demonstrate the graph algorithms and live streaming data in Spark

CO6: Analyse and implement different frame work tools by taking sample data sets.

INDUSTRY 5.0

COURSE INFORMATION:

SCHOOL: School of Engineering	**COURSE TYPE:** L-T-P
NAME: Industry 5.0	**COURSE CREDIT:** 04[3-1-0]
DEPARTMENT: Computer Science	**CATEGORY:** PCC
CODE: PGCSE302	**SEMESTER:** 3^{rd}

THEORY

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	4	17%
Module-II: Search Engine Optimization (SEO)	4	17%
Module-III: Social Media Marketing (SMM)	4	16%
Module-IV: Content Marketing	4	16%
Module-V: Online Advertising	4	17%
Module-VI: Email Marketing and Mobile Marketing	4	17%

SYLLABUS OUTLINE:

Module I: Overview: [4L]

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): [4L]

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): [4L]

About Social Media Marketing, Different types of Social Media Marketing

Module IV: Content Marketing: [4L]

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

Module V: Online Advertising: [4L]

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: [4L]

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.

- ***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

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	-	1	-	-	-	-	-	2
CO2	2	2	2	-	-	-	-	2	-	-	-	1
CO3	2	2	2	-	2	-	-	-	-	-	-	-
CO4	1	1	1	1	1	1	-	-	-	1	-	1
CO5	1	1	1	1	1	1	-	-	-	1	-	-
CO6	1	2	1	1	-	1	1	1	-	-	-	1
Avg	1.6	1.6	1.16	0.5	0.66	0.66	0.16	0.5	0.16	0.33		0.83

Highly Correlated: **3**
Moderately Correlated: **2**
Slightly Correlated: **1**

Course Learning Outcome: (CO)

CO1: Interpret Digital Marketing preliminaries.

CO2: Build effective Digital Marketing strategies for different products and services.

CO3: Make appropriate use of varied Digital Marketing Platforms like Email, Facebook, Twitter, YouTube, Pinterest, etc. as per given scenario.

CO4: Apply and analyze the concept of Search Engine Optimization (SEO), SEM and Mobile Marketing to given scenarios.

CO5: Analyze specific trends using Google Analytics.

CO6: Create effective Display Ads and Search Engine Advertising

SEMESTER-IV

Sl No	Course Title	Code	Category	Credit	Type			
					L	T	P	S
1	Dissertation/ Research Project		CC	16	0	0	0	32
2	Grand Viva		AEC	2	2	0	0	0
				Total Credit=18				