

SYLLABUS

**DEPARTMENT
OF
MASTER OF COMPUTER APPLICATIONS**

2026

MITE



Invent Solutions

**MANGALORE INSTITUTE OF
TECHNOLOGY & ENGINEERING**



MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)

Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

Institute Vision

*“To attain perfection in providing **Globally Competitive Quality Education** to all our Students and also benefit the global community by using our strength in **Research and Development**”*

Institute Mission

*“To establish world class educational institutions in their respective domains, which shall be **Centers of Excellence** in their stated and implied sense. To achieve this objective we dedicate ourselves to meet the challenges of becoming **Visionary and Realistic, Sensitive and Demanding, Innovative and Practical, Theoretical and Pragmatic; ALL at the same time**”*

Department Vision

To attain global recognition as a leader in imparting quality education in the field of computer applications by fostering an inclusive and dynamic learning environment that nurtures research skills, creativity, innovation, and critical thinking ability of the students.

Department Mission

- *To deliver modern multi-disciplinary technological skills in the field of computer applications.*
- *To cultivate a research culture among the faculty and students to the betterment of the human society at large.*
- *To make students cognizant of the importance of ethical and moral values while creating futuristic software applications.*

Program Educational Objectives (PEOs)

- *Design effective computing systems that are technically viable and fulfilling the requirements of customers.*
- *Develop a professional career to contribute towards interdisciplinary academic and research projects.*
- *Acquire and apply the knowledge of existing tools and technologies from time to time to industry-oriented and societal problems.*

Program Specific Outcomes (PSOs)

- *Apply the principles of Software Engineering and Project management to design, develop and maintain software applications using Programming languages and frameworks.*
- *Apply computational techniques and algorithms to solve real-world problems in artificial intelligence and cybersecurity environments.*

LIST OF COURSES

I/II SEMESTER			
Sl. No	Course Code	Course Title	Sem
PROFESSIONAL CORE (PC)			
1	26MCPC501	Probability & Statistics	I
2	26MCPC502	Operating Systems	I
3	26MCPC503	Data Structures	I
4	26MCPC504	Modern Software Engineering	I
5	26MCPC505	Fundamentals of Emerging Technologies	I
6	26MCPC506	Problem Solving using Python	I
7	26MCPC507	Data Structures Lab	I
8	26MCPC511	Essentials of Cloud Computing	II
9	26MCPC512	Embedded Systems and Internet of Things	II
10	26MCPC513	Programming in Java	II
11	26MCPC514	Design and Analysis of Algorithms	II
12	26MCPC516	Machine Learning	II
13	26MCPC517	Database Management System	II
HUMANITIES (HM)			
14	26MCNM508	Professional Communication and Ethics	I
15	26MCHM515	Research Methodology & IPR	II

I Semester (2026 Scheme): Department of Master of Computer Applications

Sl. No.	Course Code	Course Title	Category	Teaching Dept.	Teaching Hours/Week				Exam Marks			Duration of Exam (SEE)	Credits
					L	T	P	S	CIE	SEE	TOTAL		
1	26MCPC501	Probability & Statistics	Professional Core	Mathematics /MCA	2	1	0	2	50	50	100	3	3
2	26MCPC502	Operating Systems	Professional Core	MCA	3	0	2	2	50	50	100	3	4
3	26MCPC503	Data Structures	Professional Core	MCA	3	0	0	3	50	50	100	3	3
4	26MCPC504	Modern Software Engineering	Professional Core	MCA	3	0	0	3	50	50	100	3	3
5	26MCPC505	Fundamentals of Emerging Technologies	Professional Core	MCA	3	0	0	3	50	50	100	3	3
6	26MCPC506	Problem Solving using Python	Professional Core	MCA	2	0	2	1	50	50	100	3	3
7	26MCPC507	Data Structures Lab	Professional Core	MCA	0	0	4	0	50	50	100	2.5	2
8	26MCNM508	Professional Communication and Ethics	Humanities	Humanities /MCA	0	0	2	0	100	-	100	-	-
	Total Credits												21



MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)

Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

II Semester (2026 Scheme): Department of Master of Computer Applications

Sl. No.	Course Code	Course Title	Category	Teaching Dept.	Teaching Hours/Week				Exam Marks			Duration of Exam (SEE)	Credits
					L	T	P	S	CIE	SEE	TOTAL		
1	26MCPC511	Essentials of Cloud Computing	Professional Core	MCA	3	0	0	3	50	50	100	3	3
2	26MCPC512	Embedded Systems and Internet of Things	Professional Core	MCA	2	0	2	2	50	50	100	3	3
3	26MCPC513	Programming in Java	Professional Core	MCA	2	0	2	0	50	50	100	3	3
4	26MCPC514	Design and Analysis of Algorithms	Professional Core	MCA	3	0	0	2	50	50	100	3	3
5	26MCHM515	Research Methodology & IPR	Humanities	MCA	2	0	0	3	50	50	100	2.5	2
6	26MCPC516	Machine Learning	Professional Core	MCA	3	0	2	2	50	50	100	3	4
7	26MCPC517	Database Management System	Professional Core	MCA	3	0	2	3	50	50	100	3	4
Total Credits												22	



MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)

Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

I SEM

PROBABILITY & STATISTICS			
Semester	I	CIE Marks	50
Course Code	26MCPC501	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:0:2	Exam Hrs	3
Total Hrs	42L	Credits	3
Course Description: This course introduces the concepts of Probability, Probability Distributions, Descriptive Statistics, Statistical Inference & Hypothesis Testing, and Stochastic Processes. It develops skills in analyzing uncertainty, summarizing and interpreting data, making statistical decisions, and modeling random phenomena. The course emphasizes mathematical and statistical techniques applicable to scientific and data-driven problems.			
Course Objectives: This course is designed to 1. Impart the fundamentals of probability theory for analyzing random phenomena 2. Provide knowledge on statistical data collection, presentation, and analysis techniques 3. Provide knowledge of sampling, estimation, and hypothesis testing 4. Impart skills on probabilistic modeling and stochastic processes			
Module 1: Probability			No. of Hrs: 8L
Introduction, Algebra of Events, Addition Theorem, Conditional Probability, Multiplication Law, Baye's Theorem Self-study Component: Additional problems on Probability. Textbook 1: 3.1 to 3.15, 4.1, 4.2			
Module 2: Probability Distribution			No. of Hrs: 9L
Discrete and Continuous Random Variable, Mathematical Expectation, Mean and Variance. Binomial, Poisson and Normal Distribution. Textbook 1: 5.1 to 5.4, 6.1 to 6.6, 8.4, 8.5, 9.2			
Module 3: Descriptive Statistics			No. of Hrs: 8L
Statistics: Collection & Presentation of Data - Graphical & Tabular Representation, Measures of Central Tendency, Dispersion, Skewness. Correlation & Regression: Scatter plot, Correlation and Regression Coefficients, Curve fitting – Linear & Non-linear. Self-study Component: Additional problems on Statistics, correlation & Regression. Textbook 1: 2.1 to 2.7, 2.12 to 2.16, 10.1 to 10.5, 11.1 to 11.3			
Module 4: Statistical Inference & Hypothesis Testing			No. of Hrs: 9L
Sampling & Inference: Population and Sample, Sampling Distributions. Hypothesis testing: t-test, z-test, Interval Estimation, Non-parametric tests, Chi-Squared test for independence of attributes & Goodness of fit. Textbook 1: 14.1 to 14.6, 16.1 to 16.3			
Module 5: Stochastic Process			No. of Hrs: 8L
Introduction to Probabilistic Modelling, Stochastic Process: Types, Markov Chains, Chapman–Kolmogorov Equations for n-step Transition Probabilities, Classification of States. Self-study Component: Additional problems on Stochastic Process.			

Textbook 2: 8.1, 9.1 to 9.3

Course Outcomes:

At the end of the course, the student will be able to

1. Illustrate the fundamental concepts of Probability, Probability Distributions, Descriptive Statistics, Statistical Inference & Hypothesis Testing, and Stochastic Processes.
2. Make use of Probability, Probability Distributions, and Stochastic Processes to solve problems on random phenomenon.
3. Solve problems on Descriptive Statistics, Statistical Inference & Hypothesis Testing.
4. Solve real-world problems using Probability, Probability Distributions, Descriptive Statistics, Statistical Inference & Hypothesis Testing, and Stochastic Process models.

Textbooks:

1. S. C. Gupta & V. K. Kapoor, “Fundamentals of Mathematical Statistics”, 12th Edition, Sultan Chand & Sons, 2020
2. Scott Miller and Donald Childers, “Probability and Random Processes”, 2nd Edition, Academic Press (Elsevier), 2012

Reference Books:

1. Sheldon Ross, “A First Course in Probability”, 10th Edition, Pearson, 2023
2. Oliver C. Ibe, “Fundamentals of Applied Probability and Random Processes”, 2nd Edition, Elsevier Academic Press, 2005
3. George Casella and Roger L. Berger, “Statistical Inference”, 1st Edition, Cengage Learning India Pvt. Ltd., 2007
4. William J. Stewart, “Probability, Markov Chains, Queues, and Simulation”, Princeton University Press, 2009

Web links:

1. Essentials of Data Science with R Software-1: Probability and Statistical Inference, Prof. Shalabh, IIT Kanpur:
<https://www.youtube.com/playlist?list=PLFW6lRTa1g80s2MWqXNg2o0haq1k14v2I>
2. Descriptive Statistics Using ‘R’, Prof. Shalabh, IIT Kanpur:
<https://www.youtube.com/playlist?list=PLFW6lRTa1g83jplIote7RuEYCwOJa-6Gz>
3. Mathematics - Regression Analysis, Dr. Soumen Maity, IIT Kharagpur:
<https://www.youtube.com/playlist?list=PLbMVogVj5nJSpj5sl-8tdKARg1lw2wEa->
4. Introduction to Stochastic Processes, Prof. Manjesh hanawal, IIT Bombay:
<https://www.youtube.com/playlist?list=PLOzRYVm0a65dFqcSScAAAtAjOMGrG7luhz>

Operating Systems			
Semester	I	CIE Marks	50
Course Code	26MCPC502	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:2:2	Exam Hrs	3
Total Hrs	64 (40L+24P)	Credits	4
Course Description: This course introduces the fundamental concepts and design principles of operating systems (OS), OS structures, and system calls. Further it covers process management, CPU scheduling, process synchronization, deadlock handling, memory management, virtual memory techniques, file systems and storage management. This course also provides practical exposure to use Linux commands and write shell scripts. Further, students simulate several OS functionalities through C programming.			
Course Objectives: This course is designed to 1. Familiarize the fundamental operating systems concepts, architecture, services, structure, and design principles. 2. Impart knowledge of process management, multithreading, interprocess communication, synchronization, deadlocks, memory management, and file-system organization. 3. Familiarize operating system algorithms and techniques for CPU scheduling, memory management, and storage management. 4. Provide practical skills in Linux-based system operations, system-level programming, resource management, and shell scripting for operating system administration and automation.			
Module 1: Introduction			No. of Hrs: 8L+6P
Overview: Computer System Organization, Computer System Architecture, Operating System Operations, Resource Management, Security and Protection, Virtualization, Distributed Systems. Operating-System Structures: Operating System Services, User and Operating System Interface, System Calls, Types of System Calls, System Programs, Operating System Design and Implementation, Operating System Structure, System Boot Process.			
Laboratory Components: 1. Demonstrate the use of Linux commands ls, pwd, cd, mkdir, cp, mv, find, chmod, ps, df, and du to perform file & directory management, modify file permissions, monitor processes, and obtain system information. 2. Write a shell script to generate a system information and resource monitoring report. The script shall display the hostname, kernel version, CPU utilization, memory utilization, disk utilization, number of running processes, and logged-in users. 3. Write a C program to demonstrate system calls like fork(), exec(), wait(), getpid(), and getppid().			
Self-study Component: Computing Environments, Free & Open-source Operating Systems.			
Textbook 1: 1.1 to 1.8, 1.10 to 1.11, 2.1 to 2.3, 2.7 to 2.9			

Module 2: Process Management and CPU Scheduling	No. of Hrs: 8L+6P
Processes: Process Concept, Process States, Process Control Block, Process Scheduling, Context Switching, Operations on Processes, Inter-process Communication. Threads and Concurrency: Overview, Multi-threading Models, Thread Libraries, Threading Issues. CPU Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms: First-Come, First-Served (FCFS), Shortest Job First (SJF), Priority, Round Robin. Laboratory Components: 1. Write a C program to create parent and child processes, demonstrate process execution, termination, process identification, and process status using appropriate system calls. 2. Develop a C program to simulate the First-Come, First-Served (FCFS) and Shortest Job First (SJF) CPU scheduling algorithms for a given set of processes with arrival time and burst time. Generate the scheduling sequence and calculate the waiting time, turnaround time, average waiting time and average turnaround time. 3. Develop a C program to simulate non-preemptive Priority and Round Robin CPU scheduling algorithms for a given set of processes and time quantum. Generate the scheduling sequence and calculate the waiting time, turnaround time, average waiting time and average turnaround time. Self-study Component: Real-Time CPU Scheduling, Processor Affinity, Multi-Processor Scheduling. Textbook 1: 3.1 to 3.4, 4.1 to 4.6, 5.1 to 5.6	
Module 3: Process Synchronization and Deadlocks	No. of Hrs: 8L+4P
Synchronization Tools: Background, Critical Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classical Synchronization Problems. Deadlocks: System Model, Deadlock Characterization, Methods for Handling Deadlocks: Deadlock Prevention, Deadlock Avoidance using the Banker's Algorithm, Deadlock Detection, and Recovery from Deadlocks. Laboratory Components: 1. Write a C program to demonstrate inter-process communication between a parent and child process using pipes. 2. Write a C program to simulate solutions to the Producer–Consumer problem using POSIX threads and semaphores. Self-study Component: Monitors Textbook 1: 6.1 to 6.7, 7.1, 8.1, 8.3 to 8.8	
Module 4: Memory Management	No. of Hrs: 8L+4P
Main Memory: Background, Address Binding, Logical and Physical Address Space, Dynamic Loading and Dynamic Linking, Swapping, Contiguous Memory Allocation, Paging, and Segmentation.	

Virtual Memory: Background, Demand Paging, Copy-on-Write, Page Replacement Algorithms: First In, First Out (FIFO), Optimal, and Least Recently Used (LRU), Allocation of Frames, and Thrashing.

Laboratory Components:

1. Develop a C program to simulate contiguous memory allocation using First Fit, Best Fit, and Worst Fit techniques. For a given set of memory blocks and process requests, display the allocation status and calculate the unused/free memory.
2. Develop a C program to simulate page replacement using the FIFO page replacement algorithm. For a given page-reference string and number of available frames, display the page replacement sequence and calculate the total number of page faults.

Textbook 1: 9.1 to 9.5, 10.1 to 10.2, 10.4 to 10.6

Module 5: File Systems and Storage Management

No. of Hrs: 8L+4P

Mass-Storage Structure: Secondary Storage Devices, Disk Structure, Disk Scheduling Algorithms including FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK.

File-System Interface: File Concept, File Access Methods, Directory Structure, Protection.

File-System Implementation: File-System Structure, File Allocation Methods, Free-Space Management, Efficiency and Performance.

Laboratory Components:

1. Write a C program to simulate the First-Come, First-Served (FCFS) and Shortest Seek Time First (SSTF) disk scheduling algorithms.
2. Develop a shell script for file management and automated backup. The script shall accept a source directory, verify its existence, create a timestamped backup, compress the backup, maintain a log of backup operations, and report the success or failure of the operation.

Textbook 1: 11.1 to 11.2, 13.1 to 13.4, 14.1 to 14.6, 15.1 to 15.6, 16.1 to 16.3, 17.1 to 17.4

Course Outcomes:

At the end of the course, the student will be able to

1. Describe the fundamental concepts, principles, structures, services and operations of operating systems.
2. Describe the mechanisms and techniques used for process communication, synchronization, deadlock handling, memory management and file-system management in operating systems.
3. Illustrate the application of different process scheduling, process synchronization, memory management and disk scheduling techniques given a snapshot of a system.
4. Write C programs to simulate OS functionalities.

Textbooks

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, “Operating System Concepts”, 10th Edition, John Wiley & Sons, Inc., 2018.
2. Richard Blum and Christine Bresnahan, “Linux Command Line and Shell Scripting Bible”, 5th Edition, Wiley India Pvt. Ltd., 2025.

Reference Books

1. Andrew S. Tanenbaum and Herbert Bos, “Modern Operating Systems”, 4th Edition, Pearson Education, 2015.
2. Sumitabha Das, “UNIX/Linux Programming: A Practical Approach Using C”, McGraw Hill Education (India) Pvt. Ltd., 2018.
3. Brian Ward, “How Linux Works: What Every Superuser Should Know”, 3rd Edition, No Starch Press, 2021.
4. William E. Shotts Jr., “The Linux Command Line: A Complete Introduction”, 2nd Edition, No Starch Press, 2019.
5. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, “Operating Systems: Three Easy Pieces”, Arpaci-Dusseau Books, 2018.

Web links:

1. Fundamentals of Operating System: <https://nesoacademy.org/cs/03-operating-system>
2. Operating Systems Course for Beginners: <https://www.youtube.com/watch?v=yK1uBHPdp30>
3. The Linux Command Line: <https://linuxcommand.org/tlcl.php>
4. Linux Learning Resources: <https://researchcomputing.princeton.edu/education/external-online-resources/linux>
5. BASH Shell Script Basics: https://www.youtube.com/watch?v=9aD59kA_P1M

Data Structures			
Semester	I	CIE Marks	50
Course Code	26MCPC503	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:0:3	Exam Hrs	3
Total Hrs	42L	Credits	3
Course Description: This course introduces the fundamental concepts of data structures, their implementation and operations. It covers linear and non-linear data structures and algorithms for various searching, sorting, and hashing techniques.			
Course Objectives: This course is designed to 1. Impart knowledge on the array concepts. 2. Impart knowledge on the concepts of stack, queue, linked list and their operations. 3. Introduce the concepts of non-linear data structures-trees and graphs. 4. Familiarize the concept of sorting, searching and hashing techniques.			
Module 1 : Arrays			No. of Hrs: 8L
Introduction, Types of data structures, Arrays: Array as Abstract Data Types (ADT), Memory representation of one-dimensional arrays and operations, Two dimensional arrays: Memory representation of two-dimensional arrays and operations, Traversing linear array, Sparse arrays, Sparse matrix, operations, Representation of multidimensional arrays, Advantages and limitations of arrays, Applications of arrays.			
Self-study component: Characteristics of array, Declaration and initialization of array, Overflow and underflow in array, Memory wastage in array. Textbook 2: 4.1 to 4.5			
Module 2: Stack and Queue			No. of Hrs: 10L
Definition, Representation of stack using array, Stack as an ADT, Stack operations, Applications: Polish and reverse polish expressions, Infix to postfix conversion, Evaluation of postfix expression, Infix to prefix, Recursion. Queue: Definition, Representation, Queue as an ADT, Queue operations, Applications of queues, Queue variants: Circular queue, Priority queue, Double ended queue.			
Self-study component: Prefix to postfix, Postfix to prefix conversion, Characteristics of stack, Applications of Stack in Undo/Redo Operations, Characteristics of queue, Multiple queues. Textbook 1: 3.1, 3.2, 3.4 Textbook 2: 6.1 to 6.4, 6.6 to 6.9, 6.17			
Module 3: Linked List			No. of Hrs: 8L
Introduction to linked lists, Representation of linked list in memory, Linked list as ADT, Singly linked list, Operations on singly linked list, Doubly linked list, Operations on doubly linked list, Circular linked list, Operations on circular linked list, Implementing stack and queue using linked list, Applications of linked list, Textbook 2: 5.1 to 5.8			

Textbook 1: 4.2, 4.3, 4.8	
Module 4: Trees and Graphs	No. of Hrs: 6L
Introduction, Binary trees, Binary trees traversal, Additional binary tree Operations, Threaded binary tree, Binary search tree: Traversal, Searching in a binary search tree, Inserting into binary search tree, Deleting from binary search tree. Introduction, Graph theory terminology, Sequential representation of graph: Adjacency matrix, Path matrix, Representation of graph: Linked representation, Operations performed on graph, Graph Traversal: Depth First Search (DFS) and Breadth First Search (BFS), Applications of graph. Self-study component: Complete binary tree, Perfect binary tree, Skewed binary tree, Balanced binary tree. Advantages and limitations of tree and graph Representations.	
Textbook 1: 5.1, 5.2, 5.4, 5.5, 5.7, 5.8, 6.1 to 6.3	
Module 5: Sorting, Searching and Hashing	No. of Hrs: 10L
Sorting and searching techniques: Bubble sort, Selection sort, Insertion sort, Sequential search, Binary search. Hashing: Hash function, Address calculation techniques, Common hashing functions, Collision resolution, Linear probing, Quadratic, Double hashing, Bucket hashing, Deletion and rehashing, Static hashing: Hash Table, Hash Function, Dynamic hashing.	
Textbook 1: 7.1 to 7.3, 8.2, 8.3	
Course Outcomes: At the end of the course, the student will be able to 1. Apply array operations for efficient data storage and manipulation 2. Make use of stack, queue, and recursion techniques to solve problem 3. Apply linked list concepts in problem-solving. 4. Make use of tree and graph traversal methods to develop solution for a given problem 5. Apply appropriate sorting, searching and hashing techniques to solve problems	
Textbooks 1. Ellis Horowitz and Sartaj Sahni, “Fundamentals of Data Structures in C”, 2nd Edition, Universities Press, 2023. 2. Seymour Lipschutz, “Data Structures Schaum's Outlines”, 2nd Edition, McGraw Hill, 2014.	
Reference Books 1. Reema Thareja, “Data Structures using C”, 3rd Edition, Oxford Press, 2012.	
Web links: 1. Understanding Binary Search Tree: https://www.scribd.com/document/646762099/Binary-Search-Tree-javatpoint 2. Arrays in C: https://www.programiz.com/c-programming/c-arrays 3. Linked List: https://www.edureka.co/blog/linked-list-in-c/ 4. Stack: www.digimat.in/nptel/courses/video/106104697/L16.html 5. Hashing: https://www.geeksforgeeks.org/dsa/hashing-data-structure/ 6. Tree and Graph: https://www.w3schools.com/dsa/dsa_theory_trees.php	

Modern Software Engineering			
Semester	I	CIE Marks	50
Course Code	26MCPC504	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:0:3	Exam Hrs	3
Total Hrs	42L	Credits	3
Course Description: This course provides a comprehensive understanding of modern software engineering practices, covering software processes, Agile methodologies, requirements engineering, software modeling, design principles, and design patterns. It introduces students to software architectures such as layered, MVC, microservices, and message-oriented architectures, along with software testing, test-driven development, refactoring, and code quality. The course also covers DevOps practices, continuous delivery, source code management, automated builds, testing, deployment, and monitoring.			
Course Objectives: This course is designed to 1. Impart knowledge of modern software development processes 2. Introduce the different software models and design principles 3. Familiarize design patterns and architectural styles 4. Impart knowledge on software testing principles and refactoring techniques 5. Impart knowledge on concepts of DevOps and its benefits			
Module 1: Software Processes and Requirements			No. of Hrs: 8L
Introduction to Software Processes, Agile Manifesto, Extreme Programming, Scrum, Kanban, Limitations of Agile methods. Introduction to Requirements Engineering, User Stories, Use Cases, Obtaining a Minimum Viable Product, A/B testing. Self-study component: Other Iterative methods: Spiral model, Unified Process model Textbook 1: 1-1.3, 2-2.1-2.6, 3-3.1-3.6			
Module 2: Models and Design Principles			No. of Hrs: 9L
Introduction to Software Models, Unified Modeling Language: Class diagrams, Package diagrams, Sequence diagrams, Activity diagrams. Introduction to Design Principles: Conceptual Integrity, Information hiding, Cohesion, Coupling, SOLID and Other Design Principles, Source code metrics. Textbook 1: 4 - 4.1- 4.6, 5 - 5.1-5.7			
Module 3: Design Patterns and Architecture			No. of Hrs: 8L
Introduction to design patterns: Factory, Singleton, Proxy, Adapter, Façade, Decorator, Strategy, Observer, Template method, Visitor. Introduction to Architecture: Layered Architecture, MVC Architecture, Microservices, Message-Oriented Architectures, Publish/Subscribe Architecture. Self-study component: Other design patterns, Limitations of design patterns, Other architectural patterns, Architectural Anti-patterns Textbook 1: 6 - 6.1 - 6.11, 7.1 - 7.6			
Module 4: Software Testing and Refactoring			No. of Hrs: 9L
Introduction to Software testing: Unit testing, Principles and Smells, Test Coverage, Testability,			

Mocks, Test-Driven development, Integration tests, End-to-End tests. Introduction to Refactoring: Refactoring examples, Refactoring practice, Automated refactorings, Code smells.

Textbook 1: 8 - 8.1 - 8.9, 9 - 9.1 - 9.5

Module 5: DevOps & Automation Testing

No. of Hrs: 8L

Introduction to DevOps and Continuous delivery, The DevOps process, DevOps Vs Software Architecture, Source Code Management, Building the Code: Jenkins build server, Jenkins plugins, Jenkins filesystem layout, Testing the Code: Manual testing, JUnit, Mocking, Test Coverage, Issue Tracking Systems, Examples.

Self-study component:

DevOps and ITIL, Release Management, The Internet of Things and DevOps

Textbook 2: 1-Page Nos. 1-5, 2-Page Nos. 9-16, 3-Page Nos. 21-35, 4-Page Nos. 40-57, 5-Page Nos. 61-80, 6-Page Nos. 83-89, 9-Page Nos. 159-193.

Course Outcomes:

At the end of the course, the student will be able to

1. Describe agile methods, requirements engineering practices, MVP development, and A/B testing techniques used in software development.
2. Apply software modeling techniques, UML diagrams, design principles, and SOLID principles while designing software.
3. Make use of appropriate software design and architectural patterns for software design.
4. Apply software testing principles and refactoring techniques to a given source code.
5. Illustrate the use of DevOps practices and tools for source code management, build, testing, and continuous delivery.

Textbooks

1. Marco Tulio Valente, “Software Engineering: A Modern Approach”, 2024
2. Joakim Verona, “Practical DevOps”, Packt Publishing, 2016

Reference Books

1. Andrew Stellman & Jennifer Greene, “Head First Agile”, 1st Edition, O’Reilly Media, 2017
2. Iam Sommerville, “Software Engineering”, 9th Edition, Pearson, 2017

Web links:

1. Software Engineering Body of Knowledge: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
2. DevOps Roadmap: <https://roadmap.sh/devops>
3. Official Git Documentation: <https://git-scm.com/docs>

Fundamentals of Emerging Technologies			
Semester	I	CIE Marks	50
Course Code	26MCPC505	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:0:3	Exam Hrs	3
Total Hrs	42L	Credits	3
Course Description: This course introduces the fundamentals of emerging technologies in digital transformation and intelligent systems. It covers introduction to Data Science, Artificial Intelligence, Machine Learning, Cybersecurity, Big Data and their applications in different domains.			
Course Objectives: This course is designed to 1. Impart knowledge on Digital Transformation and Intelligent Technologies 2. Introduce fundamental concepts of Data Science and Visualization 3. Provide knowledge on fundamentals of Artificial Intelligence and Machine Learning 4. Familiarize basics of Network Security and Cybersecurity 5. Impart knowledge on Big Data and its applications			
Module 1: Digital Transformation and Intelligent Technologies			No. of Hrs: 8L
Digital Transformation, Digital Transformation Impact, Value Drivers, Digital Implementation, Methodology, Emerging Technology Heatmap, Emerging Digital Themes by Industry. Intelligent Technologies: Robotic Process Automation (RPA), Automation Capabilities, Enterprise RPA Architecture, Blockchain overview & architecture, Smart Contracts in Blockchain, Drones, Drones as Data Collector, Risk Considerations, Drone Applications, Augmented Reality, Virtual Reality and its applications.			
Textbook 1: Page No. 21 to 47, 87 to 98, 231 to 247, 267 to 278, 282 to 284.			
Module 2: Data Science Fundamentals			No. of Hrs: 8L
Introduction to Data Science, Data Science Stages, Data Science Ecosystem, Tools Used in Data Science, Data Science Workflow, Automated Methods for Data Collection, Sources of Data, Applications of Data Science. Data Analytics: Introduction, Types of statistical Data, Application. Data Visualization: Importance of Data Visualization, Conventional Data Visualization Methods, Case Study, Applications of Data Science technologies for Visualization.			
Self-study component: Data Wrangling, Data Lake, Data Warehouse, Retinal Variables in data visualization.			
Textbook 2: 1.1 to 1.10			
Textbook 3: 3.1 to 3.4, 4.1-4.7, 5.1			
Module 3: Artificial Intelligence and Machine Learning			No. of Hrs: 8L
Introduction to Artificial Intelligence, AI Capabilities, Enterprise AI Architecture, AI Agents: Agents and Environments, Good Behavior: The concept of rationality, Nature of environments, The Structure of Agents, AI Opportunities by Industry, Use Cases. Machine Learning Introduction, Types of Machine Learning, Applications of Machine Learning, Tools in Machine Learning.			
Self-study component: Risk and Benefits of AI, Difference between Human Learning and Machine Learning, Four-step process of machine learning.			
Textbook 1: Page No. 173 to 217			
Textbook 4: 2.1 to 2.4			

Textbook 5: 1.4, 1.5, 1.7, 2.1 to 2.4	
Module 4: Cybersecurity and Network Security	No. of Hrs: 10L
Introduction to cybersecurity, Need and Importance, CIA Triad, Reasons for cybercrimes, Types of cybercrimes, Types of security: Application security, Endpoint security, Mobile security, Data security. Information and Network Security Concepts: Security Attacks, Security Services, Security Mechanisms, Network Security, A Trust Model, Mitigation strategies in network security. Self-study component: Zero-day attack, Disaster Recovery, Security Standards.	
Textbook 6: 2.1 to 2.4, 2.7, 3.4, 3.5, 3.7, 4.2	
Textbook 7: 1.1 to 1.5, 1.7, 1.8	
Module 5: Big Data	No. of Hrs: 8L
Need of Big Data: Big Data characteristics, Big Data Types, Big Data Classification, Big Data handling Techniques, Designing Data Architecture, Data Sources, Data Quality, Data Pre-Processing, Data storage and analysis, Big Data applications and case studies.	
Textbook 8: 1.1 to 1.7	
Course Outcomes: At the end of the course, the student will be able to 1. Describe the concepts of digital transformation and intelligent technologies 2. Explain the fundamentals of data science and data visualization techniques 3. Explain the fundamental concepts of Artificial Intelligence and Machine Learning 4. Describe network security threats and security mechanisms 5. Describe big data analytics concepts and their applications in real world domains	
Textbooks 1. Anup Maheshwari, “Digital Transformation: Building Intelligent Enterprises”, 1st Edition, John Wiley & Sons, Inc., 2019. 2. Sandhya Arora and Latesh Malik, “Data Science and Analytics with Python”, 1st Edition, Khanna Book Publishing Co. Pvt. Ltd., 2023. 3. V.K. Jain, “Data Science and Analytics (with Python, R and SPSS Programming)”, 2nd Edition, Khanna Book Publishing Co. Pvt. Ltd., 2019. 4. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, 4th Edition, Pearson Education, 2022. 5. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, “Machine Learning”, 1st Edition, 2018, Pearson. 6. Anand Shinde, “Introduction to Cyber Security Guide to the World of Cyber Security”, 2021, Notion press 7. William Stallings, "Cryptography and Network Security: Principles and Practice", 8th Edition, Pearson, 2023. 8. Raj Kamal, Preeti Saxena “Big Data Analytics” Mc Graw Hill, 2019.	
Reference Books 1. Foster Provost and Tom Fawcett, “Data Science for Business”, 2nd Edition, O'Reilly Media, 2023.	

2. Ethem Alpaydin, “Introduction to Machine Learning”, 4th Edition, MIT Press, 2020.
3. Nina Godbole and Sunit Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, 2nd Edition, Wiley India, 2021.

Web links:

1. Introduction to Data Science: <https://www.netacad.com/courses/introduction-data-science?courseLang=en-US>
2. Foundations of Artificial Intelligence and Machine Learning: <https://skillsbuild.org/>
3. Intelligent Agents and Agent Architectures: <https://aima.cs.berkeley.edu/>
4. NIST Cybersecurity Framework (CSF 2.0): <https://www.nist.gov/cyberframework>
5. Applications of Big Data: <https://www.theknowledgeacademy.com/blog/big-data-applications/>

Problem Solving using Python			
Semester	I	CIE Marks	50
Course Code	26MCPC506	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	2:0:2:1	Exam Hrs	03
Total Hrs	50 (26L + 24P)	Credits	03
Course Description: This course introduces the fundamental concepts and programming constructs of Python. It covers Python programming fundamentals, control structures, functions, strings, collections, file handling, and exception handling. The course also emphasizes object-oriented programming concepts for developing modular, reusable, and reliable applications. It provides practical exposure to data analysis and visualization using Matplotlib and Seaborn, enabling students to interpret and present data effectively.			
Course Objectives: This course is designed to 1. Familiarize with python programming fundamentals for problem solving 2. Impart knowledge of control structures and modular programming in Python 3. Provide knowledge of data structures in python 4. Impart knowledge on Object Oriented Programming using Python 5. Familiarize the use of visualization tools in Python			
Module 1: Python Fundamentals			No. of Hrs: 4L+4P
Features, Literal Constants, Variables, Data Types, Input operations, Comments, Reserved words, Indentation, Operators and expressions, Type Conversion.			
Laboratory Components: 1. Write a program to accept a candidate's name, age, qualification, and percentage, and check admission eligibility based on age and percentage. 2. Write a Python program to accept two numbers and develop a simple calculator to perform addition, subtraction, multiplication, division, and modulus operations. 3. Write a Python program to accept the basic salary of an employee, calculate and display the HRA, DA, and gross salary, where HRA is 20% and DA is 10% of the basic salary. 4. Write a Python program to accept the temperature in Celsius, convert it to Fahrenheit and Kelvin, and display the corresponding values.			
Textbook 1 : 3.1 to 3.13			
Module 2: Control Statements and Functions			No. of Hrs: 6L+4P
Control statements: Conditional branching statements, Basic loop structures, Nested loops, The break statement, The continue statement, The pass statement, The else statement used with loops Functions: definition, Function call, Variable scope and lifetime, The return statement, Fruitful functions, Lambda Functions, Recursive function.			
Laboratory Components:			

1. Write a Python program that accepts a positive integer n from the user and performs the following operations:
 - a) Display all numbers from 1 to n that are divisible by 3.
 - b) Check whether n is even or odd and display the result.
2. Write a menu-driven Python program using user-defined functions to perform the following operations:
 - a) Generate the Fibonacci series for a given number of terms using a non-recursive function.
 - b) Find the factorial of a given number using a recursive function.
3. Write a Python program using a lambda function to find the square of a given number.

Textbook 1 : 4.2-4.8, 5.2-5.7, 5.11

Module 3: String, Collections, Files , Exception handling

No. of Hrs: 6L+6P

Strings: Concatenation, Appending and multiplication, String formatting operators, Built-in string methods, Slice operation, Comparing strings

Collections: List, Tuples, Dictionaries, Set

Files: Types, Opening and closing, Reading and Writing , File positions, Renaming and Deleting.

Exception Handling: Introduction to errors and exceptions, Handling exceptions, Multiple exception, User defined exception, The finally block.

Laboratory Components:

4. Write a program to perform string operations such as concatenation, slicing, formatting, and palindrome checking.
5. Write a program to manage student records using lists, tuples, and dictionaries.
6. Write a program to read and write text files and handle exceptions during file operations.

Self-study component: Handling CSV files, and JSON files using Python for data storage and processing.

Textbook 1 : 6.2, 6.4 to 6.6, 6.9, 7.2 to 7.6, 8.4, 8.6, 12.1 to 12.4, 12.10 to 12.11

Module 4: Object Oriented Programming with Python

No. of Hrs: 6L+6P

Classes and Objects, Class method and SELF argument, The init() method, Public and private data members, Private methods, Calling a class method from another class method, Build in functions, Build-in class attributes, Garbage collection, Class methods, Static methods, Inheriting classes, Types of inheritance, Containership, Abstract classes and interfaces.

Laboratory Components:

7. Write a program to implement a Student Management System using classes, objects, constructors, and methods.

8. Write a program to implement an Employee Management System using inheritance and encapsulation.
9. Write a program to demonstrate polymorphism using method overriding and abstract classes.

Textbook 1 : 9.1 to 9.3, 9.5, 9.8 to 9.11 to 9.15, 10.2 to 10.5

Module 5: Data Visualization

No. of Hrs: 4L+4P

Introduction to Matplotlib, Simple Line Plots, Simple Scatter Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binning and Density, Customizing Plot Legends and Color Bar.

Laboratory Components:

10. Use the Matplotlib library to create and customize Line Charts, Bar Charts, and Pie Charts using a Student Performance dataset.
11. Apply scatter plots, histograms, and box plots to explore patterns and distributions in the Student Performance dataset.
12. Use the Matplotlib library to create error bars, contour plots, and color bars for a Student Performance dataset.

Self-study component: Visualization with Seaborn.

Textbook 3 : Page Nos. 217-322

Course Outcomes:

At the end of the course, the student will be able to

- 1: Describe the fundamental features and constructs of Python programming.
- 2: Apply control structures and functions for problem solving in Python.
- 3: Apply strings, collections, file handling, and exception handling to solve programming problems.
- 4: Apply paradigm for problem solving in python.
- 5: Represent data using python visualization tool.

Textbooks

1. Reema Thareja, "Python Programming using problem solving approach", 2nd Edition, Oxford University Press, 2024.
2. Jake VanderPlas, "Python Data Science Handbook", 2nd Edition, O'Reilly Media, 2022.

Reference Books

1. Mark Lutz, "Programming Python", 4th Edition, O'Reilly Media, 2011.
2. Mark Lutz, "Learning Python", 5th Edition, O'Reilly Media, 2013.
3. Magnus Lie Hetland, "Beginning Python: From Novice to Professional", 1st Edition, Après, 2009.
4. Wes McKinney, "Python for Data Analysis", 3rd Edition, O'Reilly Media, 2022.
5. Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", 1st Edition, O'Reilly Media, 2016.

Web links:

1. Python for Data Science (NPTEL): <https://nptel.ac.in/courses/106106212>
2. Python for Everybody (Coursera): <https://www.coursera.org/specializations/python>
3. Official Python Documentation: <https://docs.python.org/3>
4. Official NumPy Documentation: <https://numpy.org/doc>
5. Matplotlib Tutorials: <https://matplotlib.org/stable/tutorials/>

Data Structures Lab			
Semester	I	CIE Marks	50
Course Code	26MCPC507	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	0:0:4:0	Exam Hrs	2.5
Total Hrs	52P	Credits	2
Course Description: This course introduces the fundamentals of data structures and algorithms using the C programming language. It covers arrays, stacks, queues, linked lists, trees, and graphs, along with searching, sorting, and hashing techniques. The course emphasizes practical implementation and efficient solutions to computational problems.			
Course Objectives: This course is designed to 1. Impart knowledge on implementing different types of arrays 2. Familiarize the implementation of stack, queue and linked list 3. Impart knowledge on graph and tree traversal techniques 4. Provide knowledge of searching, sorting and hashing techniques			
Programs on Arrays			No. of Hrs: 10P
1. Write a C program to find the largest and smallest elements in an array. 2. Write a C program to calculate the sum and average of all elements in an array. 3. Write a C program to insert a new element into an array at the desired location and print the resultant array. 4. Write a C program to delete an element from an array at a given position and display the array before and after deletion. 5. Write a C program to remove all duplicate elements from an array and display the resultant array. 6. Write a C program to add two matrices. 7. Write a C program to find the largest and smallest element in a matrix.			
Programs on Stack and Queue			No. of Hrs: 10P
1. Write a C program to implement stack with the following operations: a) Push an element on to stack. b) Pop an element from the stack c) Display 2. Write a C program to evaluate a given postfix expression using a stack. 3. Write a C program to perform the following using recursion: a) Find the Factorial of a given number. b) Generate the Fibonacci series up to n terms. 4. Write a C program to demonstrate a working of a simple queue of integers using an array. Perform the following operations: a) Insert b) Delete c) Display 5. Write a C program to demonstrate the working of a Circular Queue of integers using an array. Perform the following operations: a) Insert b) Delete c) Display			
Programs on Linked List			No. of Hrs: 12P
1. Write a C program to simulate the working of a singly linked list with the following operations: a) Insert b) Delete c) Search d) Display 2. Write a C program to implement Stack using a Linked List. 3. Write a C program to implement Queue using a Linked List.			

4. A college wishes to maintain the attendance records of students attending a laboratory session. Each student record should contain the USN, Student Name, and Attendance Status (Present/Absent). As students may join or leave the course during the semester, the attendance records need to be updated dynamically. Develop a C program to manage these records by allowing the addition of new student records, displaying the complete attendance list, searching for a student using the USN, removing the record of a student who drops the course, and determining the total number of students marked as Present.
5. Write a C program to implement a Doubly Linked List with the following operations:
a) Insert b) Delete c) Search d) Display

Programs on Trees and Graphs

No. of Hrs: 10P

1. Write a C program to traverse a given binary tree using In-order, Post-order, Pre-order techniques.
2. Write a C program to construct a Binary Tree and insert, delete and search for a given element.
3. Write a C program to find the minimum and maximum elements in a Binary Search Tree.
4. Write a C program to perform Breadth First Search (BFS) and Depth First Search (DFS) traversal of a graph starting from a given source vertex.
5. Write a C program to find all connected components in an undirected graph.

Programs on Sorting, Searching and Hashing

No. of Hrs: 10P

1. Write a C program to implement the following search techniques: a) Linear Search b) Binary Search
2. Write a C program to sort the given elements in ascending order using Bubble Sort technique.
3. Write a C program to sort the given elements in descending order using Selection Sort technique.
4. Write a C program to sort the given elements in ascending order using Insertion Sort in ascending order.
5. Write a program to implement a hash table using following collision resolution techniques a) Linear Probing b) Quadratic Probing

Course Outcomes:

At the end of the course, the student will be able to

1. Perform various operations on single and multi-dimensional arrays
2. Utilize stacks and queues to solve problems
3. Solve problems using linked list
4. Use tree and graph traversal techniques to solve given problems
5. Apply appropriate sorting, searching and hashing techniques to solve problems

Textbooks

1. Ellis Horowitz and Sartaj Sahni, “Fundamentals of Data Structures in C”, 2nd Edition, Universities Press, 2023.
2. Seymour Lipschutz, “Data Structures Schaum's Outlines”, 2nd Edition, McGraw Hill, 2014.

Reference Books

1. Reema Thareja, “Data Structures using C”, 3rd Edition, Oxford Press, 2012.

Web links:

1. Arrays in C: <https://www.programiz.com/c-programming/c-arrays>
2. Stack in C: <https://www.geeksforgeeks.org/c/implement-stack-in-c/>
3. Implement Queue: <https://www.digitalocean.com/community/tutorials/queue-in-c>
4. Tree Traversal: <https://prepinsta.com/data-structures-algorithms/inorder-postorder-preorder/>
5. Graph Traversal: <https://www.w3resource.com/c-programming-exercises/graph/index.php>
6. Hashing Techniques: <https://www.geeksforgeeks.org/dsa/hashing-data-structure/>

Professional Communication and Ethics			
Semester	I	CIE Marks	100
Course Code	26MCNM508	SEE Marks	-
Teaching Hours/Week (L:T:P:S)	0:0:2:0	Exam Hrs	-
Total Hours	26P	Credits	-
Course Description: Professional Communication and Ethics equips students with essential communication and interpersonal skills required for effective participation in academic, social, and professional environments. The course covers the fundamentals and processes of communication, barriers and remedies, active listening, verbal and non-verbal communication, group discussions, and job interviews. It develops professional writing skills through business letters, résumés, reports, emails, blogs, and cover letters. Students also learn presentation techniques, business etiquette, formal interactions, and conflict management through communication. The course aims to build confidence, clarity, professionalism, and effective interpersonal skills for successful workplace and professional interactions.			
Course Objectives: This course is designed to 1. Impart knowledge on the fundamental concepts of interpersonal and professional communication. 2. Provide knowledge on active listening with focus on content, purpose, ideas and tone. 3. Facilitate fluent speaking skills in social, academic and professional situations. 4. Disseminate effective writing skills in business, technology and academic arenas. 5. Instil essential social and professional skills necessary to navigate various interpersonal interactions effectively			
Module 1: Fundamentals of Communication			No. of Hrs: 5
Introduction, Communication—An Overview, Process of Communication, Purpose of Professional Communication, Different Forms of Communication, Communication Network in an Organization, Barriers to Communication, Some Remedies. Textbook 1: Ch 1			
Module 2: Listening Skills			No. of Hrs: 5
Developing Effective Listening Skills: Listening Is an Art, Listening vs Hearing, Poor Listening vs Effective Listening, Advantages of Good Listening, Process of Listening, Types of Listening, Intensive Listening vs Extensive Listening, Barriers to Effective Listening, Five Steps to Active Listening, Techniques for Effective Listening. Textbook 1: Ch 3			
Module 3: Speaking Skills			No. of Hrs: 5
Non-verbal Communication: Body Language, Paralinguistic Features, Proxemics/Space Distance. Group Discussions: Introduction, Difference Between GD and Debate, Personality Traits to be Evaluated, Dynamics of Group Behaviour/Group Etiquette and Mannerisms. Job Interviews: Introduction, Process, Stages in Job Interviews, Types, Desirable Qualities, Preparation, Using Proper Verbal and Non-verbal Cues, Exhibiting Confidence. Textbook1: Ch 4			
Module 4: Writing Skills			No. of Hrs: 5
Business Letters: Elements of Style, Types of Business Letters: Acknowledgement Letter, Letter of			

Recommendation, Appreciation Letter, Acceptance Letter, Apology Letter, Complaint Letter, Adjustment Letter, Inquiry Letter, Permission Letter, Invitation Letter, Rejection Letter, Order Letter, Application Letters and Cover Letters, Sales Letters.

Résumé Preparation, Business Reports: Structure of Formal Reports, Email and Blog Writing.

Textbook1: Ch 6

Module 5: Effective Presentations and Etiquettes

No. of Hrs: 6

Presentation: Essential Characteristics of a Good Presentation, preparing a presentation, delivering the presentation,

Business Etiquettes: Introductions, Formal Dress Code, Formal Dining Etiquettes, Conflict management through communication.

Textbook 2 – Ch 14,15,16

COURSE OUTCOMES:

At the end of the course, the student will be able to:

1. Show how to eliminate barriers and use verbal/non-verbal cues at social and workplace situations.
2. Outline listening strategies to comprehend wide-ranging vocabulary, grammatical structures, tone and pronunciation.
3. Interpret effective speaking skills at social, academic and business situations.
4. Interpret effective writing skills for drafting academic, business and technical documents.
5. Outline the necessary skills and knowledge to navigate social and professional interactions with confidence and grace.

Textbooks

1. Sanjay Kumar & Pushp Lata, "Communication Skills with CD". 1st Edition, New Delhi: Oxford University Press, 2018
2. P. D. Chaturvedi, Mukesh Chaturvedi, "Business Communication", 1st Edition, Pearson, 2012

Reference Books

1. Hemphill, P.D., McCormick, D. W., & Hemphill, R. D. *Business Communication with writing improvement exercises*. Upper Saddle River, NJ: Prentice Hall, 2001.
2. Raman, M., & Sharma, S. 2016. *Technical Communication: Principles and practice*. New Delhi: Oxford University Press.
3. Kaul, A. *Effective Business Communication*. Place of publication not identified: Prentice-Hall of India. 2015.
4. Rizvi, A. M. *Effective Technical Communication: A guide for Scientists and Engineers*. New Delhi: Tata McGraw Hill, 2010.

Web Links:

1. Professional Communication Skills: <https://www.onestopenglish.com/adults/business-and-esp/esp-lesson-plans/professional-communication-skills>.
2. Effective Communication: <https://communicationprogram.wharton.upenn.edu/library/>
3. Professional Communication: <https://www.futurelearn.com/courses/effective-communication-skills-for-professionals>
4. Business Etiquette: <https://www.youtube.com/watch?v=15XC6R80wb4>



MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

(A Unit of Rajalaxmi Education Trust®, Mangalore)

Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

Accredited by NAAC with A+ Grade & ISO 9001:2015 Certified Institution

II SEM

Essentials of Cloud Computing			
Semester	II	CIE Marks	50
Course Code	26MCPC511	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:0:3	Exam Hrs	3
Total Hrs	42L	Credits	3
Course Description: This course introduces the fundamental concepts, technologies, and applications of cloud computing. It covers the evolution, characteristics, and benefits of cloud computing, along with various cloud service and deployment models. The course explores cloud architecture, virtualization, containerization, and cloud infrastructure, providing an understanding of how cloud resources and services are designed, provisioned, and managed. It further addresses essential aspects of cloud security, governance, privacy, and compliance, and introduces emerging technologies and trends such as server less computing, edge computing, multi-cloud, and cloud-native technologies. The course provides a foundation for understanding and utilizing cloud platforms for modern computing applications.			
Course Objectives: This course is designed to 1. Familiarize cloud technologies, cloud service models, and deployment models 2. Impart knowledge on virtualization, containerization, and their significance in cloud computing 3. Introduce cloud compute, storage, networking, and resource provisioning concepts 4. Provide knowledge on cloud security challenges and measures 5. Familiarize with cloud governance, compliance, and emerging cloud computing technologies			
Module 1: Introduction			No. of Hrs: 7L
Evolution of Cloud Computing: Parallel Computing Vs Distributed Computing Vs Cloud Computing, Characteristics of Cloud Computing, Cloud Service Models: SAAS, PAAS, IAAS, XAAS, Types of Clouds: Public, Private, Hybrid, Community Clouds, Cloud Architecture and Cloud Service Providers. Textbook 1-1.1,2.1 to 2.4,4.2 to 4.4			
Module 2: Virtualization and Containers.			No. of Hrs: 09L
Virtualization: Need for Virtualization, Characteristics of Virtualized Environments, Virtualization and Cloud Computing, Types of Virtualizations, Hypervisors, Types of Hypervisors, Advantages and Limitations of Virtualization, Virtual Machine Provisioning and Migration. Containers: Virtual Machines vs Containers, Building Blocks of Containers, Container Architecture, Container Orchestration, Introduction to Kubernetes, Kubernetes Architecture, Deployments and Services. Self-Study Component: VM on AWS Vs Azure Vs GCP. Textbook 1 : 3.2 to 3.6, 13.2 to 13.4			
Module 3: Cloud Infrastructure			No. of Hrs: 08L
Compute: Load balancing, Logical Network Perimeter, Virtual Server, Cloud Usage Monitor Cloud Storage: Data Replication, Distributed Storage, Storage as a service.			

Networks: Content Delivery Networks (CDN), Virtual Private Cloud (VPC), Regions and Availability Zones, Resource provisioning

Self-Study Component: Storage and Server-less Computing in Azure/AWS/GCP

Textbook 2 7.1 to 7.7

Textbook 5- 15.1,15.2,15.4,15.5,8.5

Module 4: Security Threats and Mitigation

No. of Hrs: 09L

Security Threats: Data Breaches, Account Hijacking, Insecure APIs, Denial-of-Service Attacks, Insider Threats, Malware & Ransomware

Mitigation: Challenges, Shared Responsibility Model, Identity and Access Management System, Authentication & Authorization, Role-based Access Control, Encryption, Key Management, Firewall, Logs and Traffic Monitoring, Intrusion Detection and Prevention Systems (IDS/IPS), Virtual Private Networks (VPN).

Self-Study Component: Cloud Data Backup and Disaster Recovery.

Textbook 2: 5.2 to 5.4, 8.1 to 8.28

Module 5: Cloud Governance & Emerging Cloud Technologies

No. of Hrs: 09L

Cloud Governance: Internal Policy Compliance, Governance, Risk, and Compliance (GRC), Auditing the Cloud for Compliance, Service Level Agreement (SLA): SLA parameters, Availability & Performance metrics, Service Level Objectives (SLOs), Quality of Service (QoS).

Emerging Cloud Technologies: AI/ML on Cloud, Edge Computing, Fog Computing, Cloud federation, Multicloud, Sustainable Cloud Computing.

Textbook 3: 8 chapter 167-194

Textbook 2:15.12, 15.13 ,14.12,14.13,9.3

Textbook 4: 10

Course Outcomes:

At the end of the course, the student will be able to:

1. Describe the fundamental concepts, service models, and deployment models in cloud computing for the significance of virtualization, container orchestration in cloud environment.
2. Articulate necessity of cloud security measures for data protection, access control and security threat mitigation mechanisms.
3. Describe emerging cloud technologies such as Edge, Fog and AI/ML on cloud.
4. Explain appropriate selection of compute, storage and networking resources to build a cloud infrastructure and significance of best cloud governing practices.
5. Build basic cloud applications using Python and AWS.

Text Books

1. Buyya, R., Vecchiola, C., & Selvi, S. T., "Mastering Cloud Computing: Powering AI, Big Data, and IoT Applications", 2nd Edition, McGraw Hill Education (India) Private Limited, 2024.
2. Erl, T., & Barceló Monroy, E., "Cloud Computing: Concepts, Technology and Architecture", 2nd Edition, Pearson Education, 2024.
3. Mather, T., Kumaraswamy, S., & Latif, S., "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", 1st Edition, O'Reilly Media, 2009.

4. Natarajan, S., & Jacob, J., “Multi-Cloud Handbook for Developers: Learn how to Design and Manage Cloud-Native Applications in AWS, Azure, GCP, and More”, 1st Edition, Packt Publishing, 2024.
5. Bhowmik, S., “Cloud Computing”, 1st Edition, Cambridge University Press, 2017

Reference Books

1. Kavis, M. J., “Architecting the Cloud: Design Decisions for Cloud Computing Service Models”, 1st Edition, Wiley, 2014.
2. Poulton, N., “Docker Deep Dive: Zero to Docker in a Single Book”, 1st Edition, Packt Publishing, 2023.
3. Hightower, K., Beda, J., & Burns, B., “Kubernetes Up & Running: Dive into the Future of Infrastructure”, 3rd Edition, Shroff Publishers, 2022.

Web links:

1. Cloud Computing Basics: <https://www.youtube.com/watch?v=64-1ymY2xaw>
2. Cloud computing tutorial for beginners: <https://www.pragimtech.com/blog/cloud/cloud-tutorial-for-beginners/>
3. Introduction to Cloud Computing: <https://www.coursera.org/learn/introduction-to-cloud>
4. Docker Containers and Kubernetes Fundamentals: <https://www.youtube.com/watch?v=kTp5xUcalw>

Embedded Systems and Internet of Things			
Semester	II	CIE Marks	50
Course Code	26MCPC512	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	2:0:2:2	Exam Hrs	3
Total Hrs	50 (26L+24P)	Credits	3
Course Description: This course introduces the fundamental concepts of Embedded Systems and Internet of Things (IoT). It covers the architecture of the ESP32, Embedded C programming, Interfacing of Peripherals, communication protocols in Embedded and IoT systems. It also emphasizes the design & development of Embedded and IoT systems for real-world applications.			
Course Objectives: This course is designed to 1. Impart fundamental knowledge on Embedded and IoT systems 2. Provide knowledge on ESP32 architecture and Embedded C programming 3. Familiarize communication protocols used in Embedded and IoT systems 4. Impart knowledge on design & development of Embedded and IoT systems for real-world applications.			
Module 1: Introduction to Embedded Systems			No. of Hrs: 5L
Application Domain of Embedded Systems, Desirable Features and General Characteristics of Embedded Systems, Model of an Embedded System, Microprocessor vs Microcontroller, Example of a Simple Embedded System, Figures of Merit for an Embedded System, Microcontroller Data Handling Capacity and Bit-Based Classification, Current Trends. Textbook 1: 1.1 to 1.9			
Module 2: Microcomputer Architecture and Embedded C Programming			No. of Hrs: 6L+6P
Microcomputer Architecture: Microcomputer Structure, Central Processing Unit, System Buses, Memory Organization, I/O Subsystem Organization, Interrupts. ESP32 Architecture and Peripherals: ESP32 overview, Features, Pinout Diagram, Architecture, Processor and Memory Organization, General-Purpose Input/Output (GPIO), Analog-to-Digital Converter (ADC), Digital-to-Analog Converter (DAC), Pulse Width Modulation (PWM), Timers, Interrupts, Wi-Fi, Bluetooth and Bluetooth Low Energy (BLE). Embedded C Programming: Introduction, Program Structure, Variables, Data Types, Operators, Control Structures, GPIO, ADC, PWM, Timers and Interrupts Programming, Serial Communication and Debugging, and Basic Programming Examples.			
Laboratory Components: 1. Interface an LED and push button with the ESP32 using GPIO pins and write an Embedded C program to read the push-button input and control the LED. The LED should turn ON when the push button is pressed and OFF when the push button is released. 2. Interface an analog potentiometer with the ESP32 ADC and write an Embedded C program to read the analog input value. Use the ADC value to control the brightness of an LED through PWM, such that the LED brightness varies according to the potentiometer value. 3. Interface and configure the hardware timer of the ESP32 and write an Embedded C program to generate a periodic event using a timer interrupt. Display the corresponding event status and count through serial communication at regular time intervals.			

Self-study Component: ESP32 interrupts and event-driven control using Embedded C.

Textbook 2: 3.1, 3.3 to 3.6

Study Material: 2.1 to 2.2

Module 3: Sensors, Actuators and Communication

No. of Hrs: 5L+6P

Sensors: Introduction, Characteristics, Classification, Types: Temperature, Pressure, Motion, Proximity, Gas, Infrared (IR), Ultrasonic, Humidity.

Actuators: Introduction, Types: Hydraulic actuators, Pneumatic actuators, Thermal actuators, Electromechanical actuators (Relay, DC and Servo Motor).

Communication Protocols: Serial Communication, Universal Asynchronous Receiver/Transmitter (UART), Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C).

Laboratory Components

1. Interface a temperature sensor with the ESP32 and write an Embedded C program to read the temperature value and display the measured temperature through the ESP32 Serial Monitor.
2. Interface an ultrasonic distance sensor and a servo motor with the ESP32. Write an Embedded C program to measure the distance and control the servo motor based on the measured distance.
3. Interface a gas sensor and relay module with the ESP32. Write an Embedded C program to detect the presence of gas and activate/deactivate the relay based on the sensor output.

Self-study Component: Text message transmission between two ESP32 boards using UART serial communication.

Textbook 3: 4.1 to 4.7

Textbook 4: 1.1 to 1.11, 2.1 to 2.6, 6.1 to 6.5

Textbook 5: 1.7

Module 4: Internet of Things

No. of Hrs: 5L+6P

Introduction, Working, Components, Advantages, Disadvantages, Different IoT scenarios, Characteristics, Applications and Challenges. Physical Design, Logical Design, Enabling Techniques, IoT levels, Architecture, Design Principles, IoT and M2M Technology, Real-world Design Constraints.

Laboratory Components:

1. Develop an IoT-based temperature monitoring application using ESP32 and write an Embedded C program to interface a temperature sensor, acquire the sensor data, and upload it to Cloud for real-time monitoring.
2. Develop an IoT-based remote LED control application using ESP32 and write an Embedded C program to control an LED remotely through a Wi-Fi-based IoT interface.
3. Develop an IoT-based security system using ESP32. Interface a PIR motion sensor with the ESP32 to detect human movement. When motion is detected, activate an LED/buzzer and transmit the motion detection status to a cloud/IoT platform through Wi-Fi for remote monitoring

Textbook 3: 1.1 to 1.9, 2.1 to 2.6, 2.8 to 2.10

Module 5: Communication Protocols in IoT	No. of Hrs: 5L+6P
Message Queuing Telemetry Transport (MQTT), Zigbee, Bluetooth/BLE, HTTP, Wi-Fi, TCP/UDP, Advanced Message Queuing Protocol (AMQP), Constrained Application Protocol (CoAP), CoAP vs MQTT, RF, IPv4 vs IPv6, 6LoWPAN. Laboratory Components: 1. Develop an IoT-based gas monitoring application using ESP32 and write an Embedded C program to interface a gas sensor, acquire the gas level, and publish the sensor data to an MQTT broker for remote monitoring. 2. Develop an IoT-based environmental monitoring application using ESP32 and write an Embedded C program to interface a DHT22 temperature and humidity sensor and transmit the measured data to a Bluetooth/BLE-enabled device. 3. Develop an IoT-based distance monitoring application using ESP32 and write an Embedded C program to interface an ultrasonic sensor, acquire the distance of an object, and transmit the measurement using the CoAP protocol. Self-study Component: Zigbee-based wireless communication for IoT sensor networks. Textbook 3: 3.1 to 3.13	
Course Outcomes: At the end of the course, the student will be able to 1. Describe the characteristics, structure, importance, and functional components of embedded systems 2. Describe microcontroller architecture, pinouts, communication protocols, and applications 3. Describe IoT architecture, communication protocols, and applications in IoT-based systems 4. Write Embedded C programs for interfacing sensors, actuators, and peripherals in microcontroller-based applications 5. Develop IoT applications using ESP32 by interfacing sensors and actuators and implementing IoT communication protocols	
Textbooks 1. L. B. Das, Embedded Systems: An Integrated Approach, 1st ed. Pearson Education India, 2012. 2. Manuel Jiménez, Rogelio Palomera, and Isidoro Couvertier, Introduction to Embedded Systems: Using Microcontrollers and the MSP430, Springer, 2014 3. Neera Batra and Sonali Goyal, IoT Fundamentals with a Practical Approach, 1st ed., CRC Press, 2025. 4. Dawoud Shenouda Dawoud and Peter Dawoud, Serial Communication Protocols and Standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX, River Publishers, 2020. 5. Yogesh Misra, Programming and Interfacing with Arduino, 1st ed., CRC Press, 2022.	
Reference Books 1. Study Material: Dr. Chandrakantha T. S. and Dr. Prashanth C. M., Microcontroller Architecture and Embedded C Programming, Module 2 Study Material, Course Code:	

26MCPC512, Semester II, Master of Computer Applications (MCA), Mangalore Institute of Technology and Engineering (MITE), 2026.

2. Frank Vahid and Tony D. Givargis, “Embedded System Design: A Unified Hardware/Software Introduction”, 1st Edition, John Wiley & Sons Inc., 2002.
3. Shibu Kizhakke Vallathai, “Introduction to Embedded Systems”, 2nd Edition, McGraw Hill Education (India) Private Limited, 2017.
4. Margolis, Michael, and Nicholas Weldin. Arduino Cookbook. 2nd Edition, 2012. O’Reilly Media, Inc.
5. Misra, Yogesh. Programming and Interfacing with Arduino. 1st Edition, 2022.
6. Arshdeep Bahga and Vijay Madisetti, “Internet of Things: A Hands-On Approach”, 1st Edition, Universities Press, 2015.
7. Olivier Hersent, David Boswarthick, and Omar Elloumi, “The Internet of Things: Key Applications and Protocols”, 2nd Edition, John Wiley & Sons, 2012.

Web links:

1. NPTEL – Embedded Systems (IIT Delhi):<https://nptel.ac.in/courses/108102045>
2. NPTEL – Industry 4.0 and IIoT (IIT Kharagpur): <https://nptel.ac.in/courses/106105195>
3. IoT Full Course (Edureka):<https://youtu.be/h0gWfVCSGQQ?si=EZOAdXc2FeFmNode>
4. Embedded System Design: A Unified Hardware/Software Introduction (Frank Vahid & Tony Givargis):<http://esd.cs.ucr.edu/>
5. Programming Embedded Systems: <https://www.programmingembeddedsystems.com/>

Programming in Java			
Semester	II	CIE Marks	50
Course Code	26MCPC513	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	2:0:2:0	Exam Hrs	03
Total Hrs	50 (26L + 24P)	Credits	03
Course Description: This course introduces the fundamental concepts and features of Java programming. It covers arrays, strings, classes, inheritance, interfaces, packages, exception handling, and multithreading. The course also enables students to develop java applications.			
Course Objectives: This course is designed to 1. Familiarize with fundamental concepts of Java programming. 2. Impart knowledge of classes, objects, and methods in Java. 3. Provide knowledge of inheritance, interfaces and packages in Java. 4. Familiarize the handling of Exceptions and multithreaded programming in Java.			
Module 1: Overview of Java			No. of Hrs: 6L+4P
Introduction to Object-Oriented Programming (OOP) – Two Paradigms, Principles of OOP. Lexical Elements: Whitespaces, Identifiers, Literals, Comments, Separators, Keywords. Datatypes, Literals, Variable declaration and initialization, Type conversion and casting. Arrays: One dimensional and multi-dimensional arrays. Operators: Arithmetic, Bitwise, Relational, Assignment, Operator Precedence. Control Statements: Selection, Iteration, Jump Statements.			
Chapter 2, 3, 4 and 5 Laboratory Components : 1. Write a Java program to find the sum of all odd numbers and even numbers from 1 to n. 2. Write a Java program to find the sum of following series up to n terms: a) $S = 1/1! + 2/2! + 3/3! + 4/4! + \dots + n/n!$ b) $S = 1/1! - 2^2/2! + 3^3/3! - 4^4/4! + 5^5/5! - \dots \pm n^n/n!$ 3. Write a menu driven Java program to store N integer elements in an array and perform the following operations: 1. Find the largest and smallest element. 2. Calculate the sum and average of elements. 3. Count the number of even and odd numbers. 4. Display the array elements in reverse order.			
Module 2: Classes and Methods			No. of Hrs: 6L+4P
Defining a Class, Declaring Objects, Assigning Object Reference Variables, Methods, Constructors, this keyword, Garbage Collection, Overloading Methods, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static and final keywords, Nested and Inner Classes, String class, Using Command-Line Arguments, Varargs: Variable-Length Arguments.			
Chapter 6 and 7			

Laboratory Components

1. Write a Java program to create a Rectangle class with length and width. Use overloaded methods to calculate the area of a rectangle and a square.
2. Write a Java program to create an outer class University with an inner class Department and display the department details.
3. Develop a Java application to validate the strength of a password. The password should meet the following requirements:
 - Minimum specified length
 - At least one uppercase letter
 - At least one lowercase letter
 - At least one digit
 - At least one special character
4. Write a Java program to generate the electricity bill. Accept Consumer Name, Consumer ID, and Units Consumed as command line arguments given :

Unit	Rate
0–100	Rs.2/unit
101–200	Rs.3/unit
Above 200	Rs.5/unit

Display the consumer details and total bill amount.

Module 3: Inheritance, Interface and Packages

No. of Hrs: 5L+4P

Inheritance: Inheritance basics, Using super keyword, Creating Multilevel Hierarchy, Method Overriding, Using Abstract Classes, Using final with Inheritance, Object class.

Interfaces: Definition, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods.

Packages: Packages and Member Access, Importing Packages.

Chapter 8 and 9

Laboratory Components

1. Write a Java program to implement multilevel inheritance using three classes: Employee, Developer, and ProjectLead. Include appropriate data members and methods in each class to store and display employee details, developer details, and project lead details.
2. Develop a Java program for Vehicle Management using multiple inheritance. Create two interfaces, Vehicle and Engine, with appropriate methods such as displayVehicleDetails() and calculateMileage() in the Vehicle interface, and startEngine() and stopEngine() in the Engine interface. Use appropriate variables such as vehicleNumber, vehicleModel, and mileage. Implement both interfaces in a single class Car.
3. Create a user-defined package named arithmetic containing a class Calculator with methods to perform addition, subtraction, multiplication, and division. Write a Java program to import the package, create an object of the Calculator class, and invoke all the methods to display the results.

Module 4:Exception Handling	No. of Hrs: 4L+6P
Exception Handling: Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Exception Subclasses. Chapter 10 Laboratory Components 1. Write a Java program to read the marks of 5 students into an integer array. Use nested try blocks to access a student's marks using a user-given index. Use throw to generate an exception if the marks are less than 0 or greater than 100. Use nested try blocks, throw, multiple catch blocks, and finally to handle invalid marks and invalid array positions. 2. Write a Java program to declare an integer array (a[]) with elements {10, 15, 5, 20, 25, 30} and evaluate the expression $x = a[i] / (a[i] - 5)$ by accessing the array elements. Use appropriate exception handling mechanisms to handle the Exceptions. Display suitable messages for the Exceptions. 3. Write a Java program to create a user-defined exception to validate a student's age (minimum age 18 years). If the age is invalid, throw the user defined exception (demonstrate exception chaining).	
Module 5 : Multi-Threaded Programming	No. of Hrs: 5L+6P
Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Inter-thread Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State. Chapter 11 Laboratory Components 1. Write a Java program to create a thread by implementing the Runnable interface and display the multiplication table of a given number. 2. Write a Java program to create threads for the following operations. a. Thread 1: Display the natural numbers from 1 to 100. b. Thread 2: Display even numbers from 2 to 100 c. Thread 3: Display odd numbers from 1 to 99. Set Thread Properties so that Thread 2 must run at beginning followed by Thread 3 and at last first Thread 1 must be executed. 3. Write a Java program to simulate deposit and withdrawal operations on a bank account by two different Threads. Use appropriate synchronization techniques to ensure that the account balance is updated correctly and race conditions are avoided. Display the account balance after each transaction and the final balance after all transactions are completed.	
Course Outcomes: At the end of the course, a student will be able to: 1. Describe fundamental concepts of Java programming. 2. Develop object-oriented Java applications. 3. Write Java programs to demonstrate exception handling mechanisms and multithreading	

techniques.
Textbooks 1. Herbert Schildt and Dr. Danny Coward, "Java: The Complete Reference", 13 th Edition, McGraw-Hill, 2024.
Reference Books 1. Marc Loy, Patrick Niemeyer and Daniel Leuck, "Learning Java: An Introduction to Real-World Programming with Java", 6 th Edition, O'Reilly Media, 2023. 2. Cay S. Horstmann, "Core Java: Fundamentals (Volume I)", 13 th Edition, Pearson, 2025. 3. P. J. Deitel and H. M. Deitel, "Java: How to Program", 6 th Edition, Prentice Hall, 2005. 4. P. Radha Krishna, "Object-Oriented Programming through Java", 1 st Edition, CRC Press, 2007. 5. Saurabh Choudhary, "Programming in Java", 6 th Edition, Sachin Malhotra, 2018.
Web links: 1. Basic Java programming : https://www.codecademy.com/learn/learn-java 2. Classes and Objects : https://www.w3schools.com/java/java_classes.asp 3. Packages in Java : https://docs.oracle.com/javase/tutorial/java/package 4. Exception Handling in Java: https://docs.oracle.com/javase/tutorial/essential/exceptions/ 5. Multi-Threaded Programming : https://wiki.luckfox.com/Core3566/Linux-Systems-Programming/5.Multithreaded-Programming/

Design and Analysis of Algorithms			
Semester	II	CIE Marks	50
Course Code	26MCPC514	SEE Marks	50
Teaching Hrs /Week (L: T:P:S)	3:0:0:2	Exam Hrs	03
Total Hrs	42L	Credits	03
Course Description: This course provides a comprehensive understanding of algorithm design and analysis, with emphasis on evaluating algorithm efficiency using time and space complexity and asymptotic notations. It covers fundamental algorithm design techniques including brute force, divide-and-conquer, decrease-and-conquer, greedy methods, dynamic programming, backtracking, and branch-and-bound for solving computational problems. The course also introduces string-matching algorithms and computational complexity, including complexity classes and NP-completeness. Through the study of various algorithmic approaches, the course enables systematic selection and application of suitable techniques for solving computational problems.			
Course Objectives: This course is designed to 1. Provide knowledge on asymptotic notations, time and space complexity 2. Impart knowledge on Divide and Conquer and Decrease and Conquer techniques 3. Familiarize Greedy approaches for algorithm design 4. Provide knowledge on Dynamic Programming principles, Backtracking, and Branch-and-Bound techniques 5. Impart knowledge of string-matching algorithms, complexity classes, and NP-completeness			
Module 1: Introduction			No. of Hrs: 08
Introduction to Algorithms: Overview of Algorithms, Need for Algorithm Efficiency, Fundamental Stages of Problem Solving, Basics of Algorithm Analysis: Basics of Algorithm Complexity, Introduction to Time complexity, Space Complexity, Analysis of Iterative Algorithms, Best, Worst and Average-case Complexity, Rate of Growth, Asymptotic Analysis.			
Text Book 1: 1.1, 1.4, 1.5, 1.6, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6			
Module 2: Basic Algorithm Design Techniques			No. of Hrs: 08
Mathematical Analysis of Recursive Algorithms: Introduction to Recurrence Equations, Formulation of Recurrence Equations, Techniques for Solving Recurrence Equations, Divide-and-conquer Recurrences: Master Theorem, Brute Force Approaches: Linear Search, Analysis of Recursion Programs, Sorting Problem, Bubble Sort, Selection Sort, Divide-and-conquer Approach: Introduction, Merge Sort, Quick Sort.			
Text Book 1: 4.1, 4.2, 4.3.1, 4.3.2, 4.3.3, 4.3.6, 4.5, 7.2, 7.2.1, 7.3, 7.3.1, 7.3.2, 8.1, 8.2, 8.3			
Module 3: Decrease-and-conquer & Greedy Algorithms			No. of Hrs: 09
Decrease-and-conquer Approach: Introduction, Decrease by Constant Method: Insertion Sort, Topological Sort, Generating Subsets, Decrease by Constant Factor Method: Binary Search. Greedy Algorithms: Introduction to the Greedy Approach, Components of Greedy Algorithms: Properties of Greedy algorithms, Coin Change Problem, Scheduling Problems: Shortest Job First Scheduling, Scheduling with Deadline, Knapsack Problem, Huffman Coding, Minimum Spanning Trees.			
Text Book 1: 9.1, 9.2.1, 9.2.2, 9.2.4, 9.3.1, 11.1- 11.4.2, 11.5, 11.8, 11.10			
Self-study Component: Generating Permutations (9.2.3), Tree Vertex Splitting Problem (11.9).			

Module 4: Dynamic Programming and Backtracking	No. of Hrs: 08
Dynamic Programming: Basics, Transitive Closure and Warshall Algorithm, Floyd-Warshall All-Pairs Shortest-path Algorithm, Bellman-Ford Algorithm, Traveling Salesperson Problem, Knapsack Problem, Optimal Binary Search Trees, Dynamic Programming Approach for Constructing Optimal Binary Search Trees, Backtracking: Basics, N-Queen Problem, Sum of Subsets, Vertex Coloring Problem. Text Book 1: 13.1, 13.6, 13.7, 13.8, 13.9, 13.11, 13.12, 14.2, 14.3, 14.4, 14.5 Self-study Component: Hamiltonian Circuit Problem (14.6)	
Module 5: Branch-and-Bound, String Algorithms and Computational Complexity	No. of Hrs: 09
Branch-and-bound Technique: Introduction, Assignment Problem, Traveling Salesperson Problem, Basics of Computational Complexity: Complexity Classes, Theory of NP-complete Problems, NP-Completeness. Example Problems for Proving NP-completeness: SAT (Boolean Satisfiability Problem) is NP-complete, CNF-SAT is NP-complete, Clique Decision Problem is NP-complete, Sum of Subsets (from 3-CNF-SAT). Text Book 1: 15.1, 15.5, 15.6, 18.5, 18.6, 18.8, 18.9 Self-study Component: Branch-and-bound Technique: Knapsack Problem (15.7)	
Course Outcomes: At the end of the course, the student will be able to CO1: Utilize algorithmic concepts and asymptotic analysis to evaluate algorithm efficiency. CO2: Make use of Divide-and-Conquer and Decrease-and-Conquer techniques to solve computational problems. CO3: Apply Greedy techniques to solve problems and compute their complexity. CO4: Apply Dynamic Programming, Backtracking, and Branch-and-Bound techniques to solve combinatorial problems. CO5: Determine the NP-completeness of given computational problems.	
Textbooks: 1. S. Sridhar, "Design and Analysis of Algorithms", 2nd Edition, Oxford University Press, 2024	
Reference Books: 1. Anany Levitin, Introduction to the Design and Analysis of Algorithms, (Indian), Pearson, 2017 2. Cormen T, Leiserson C., Rivest R. and Edition, MIT Press, 2009 3. Horowitz E., Sahni S. and Rajasekaran S, "Fundamentals of Computer Algorithms" 2nd Edition, University Press (India) Pvt. Ltd, 2008 4. Dave P. H. and Dave H. B, "Design and Analysis of Algorithm Education India, 2013 5. Tamassia R. and Goodrich M. T, "Algorithm Design and Application Wiley, 2014	
Web Links: 1. Design and Analysis of Algorithms: https://www.codechef.com/learn/course/college-design-analysis-algorithms 2. VisuAlgo – Algorithm Visualization: https://visualgo.net/ 3. Design and Analysis of Algorithms CS161: https://online.stanford.edu/courses/cs161-design-and-analysis-algorithms 4. Design and Analysis of Algorithms(OpenCourseWare): https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/	

Research Methodology and IPR			
Semester	II	CIE Marks	50
Course Code	26MCHM515	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	2:0:0:0	Exam Hrs	2.5
Total Hrs	26L	Credits	02
Course Description: This course develops skills in reviewing literature, identifying research problems, formulating research methodologies and communicating research findings effectively. The course emphasizes research ethics, academic integrity and citation practices. It also provides an understanding of intellectual property rights, prior-art search, copyrights, trademarks, and patent filing procedures. Practical activities enable students to apply research methods, scholarly tools, data analysis and research communication.			
Course Objectives: This course is designed to 1. Provide knowledge on how to define and approach a research problem 2. Familiarize with systematic literature survey 3. Impart skills to develop theoretical/conceptual frameworks and write effective review articles 4. Familiarize with Intellectual Property Rights			
Module 1: Foundations of Research and Literature Review			No. of Hrs: 07
Introduction: Meaning of Research, Objectives of Research, Motivation in Research, Types of Research, Research Approaches, Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is Done, Research Process, Criteria of Good Research.			
Reviewing the Literature: New and Existing Knowledge, Analysis, Bibliographic Databases, Effective Search Strategies, Systematic Literature Review, Analysis and Synthesis of Literature, Identification of Research Gaps, Development of Theoretical and Conceptual Frameworks.			
Textbook 1 – 1, Textbook 2 – 3			
Module 2: Research Problem Formulation and Design			No. of Hrs: 05
Research Problem, Selecting the Problem, Necessity of Defining the Problem and Techniques Involved in defining a Problem.			
Research Design: Meaning of Research Design, need for Research Design, features of a Good Design, Important Concepts Relating to Research Design, Different Research Designs, Basic Principles of Experimental Designs.			
Textbook 1 – 2, 3			
Module 3: Report Writing			No. of Hrs: 05
Significance of report writing, Different steps in writing a report, Layout of the research report, Types of reports, Mechanics of writing a research report and Precautions for writing research reports.			

Textbook 1 – 14	
Module 4: Intellectual Property Rights (IPR)	No. of Hrs: 05
Introduction: Role of IP in the Economic and Cultural Development of the Society, IP Governance, IP as a Global Indicator of Innovation, Origin of IP History of IP in India- Copyrights, Patents, Trademarks, Geographical Indications and Trade secrets.	
Textbook 3 – 4, 5	
Module 5: Process of Patenting	No. of Hrs: 04
Prior Art Search, Choice of Application to be Filed, Patent Application Forms, Jurisdiction of Filing Patent Application, Publication, Pre-grant Opposition, Examination, Grant of a Patent, Validity of Patent Protection, Post-grant Opposition and Commercialisation of a Patent.	
Textbook 3 – 6	
Course Outcomes: At the end of the course, the student will be able to: 1. Describe the basic research principles and methodologies 2. Differentiate theoretical and conceptual frameworks through Literature Review, Technical Reading, Attributions and Citations 3. Write effective technical reports and research papers. 4. Describe Intellectual Property concepts to address issues related to technology innovation.	
Textbooks: 1. C. R. Kothari, “Research Methodology, Methods and Techniques”, 4th edition, Gaurav Garg New Age International, 2018 2. Ranjit Kumar, “Research Methodology a step-by- step guide for beginners”, 5th Edition, SAGE Publications, 2019. 3. Dr. Santosh M Nejar, Dr. Harish Bendigeri “Research Methodology and Intellectual Property Rights”, 4th Edition, 2023.	
Reference Books: 1. David V. Thiel “Research Methods for Engineers”, 1st Edition, Cambridge University Press, 2014. 2. N. K. Acharya “Intellectual Property Rights”, 6th Edition Asia Law House, 2013.	
Web Links: 1. Introduction to Research Methodology: https://www.enago.com/academy/choose-best-research-methodology/ 2. Research Methodologies: https://library.tiffin.edu/researchmethodologies/whatareresearchmethodologies 3. Reference Management tool: https://desktop-download.mendeley.com/download/Getting_Started_Guide.pdf 4. IPR and Patents: https://www.editage.com/insights/patents-101-what-are-patents-and-how-do-you-acquire-them?refer=scroll-to-2-article&refer-type=article 5. WIPO PATENTSCOPE: https://patentscope.wipo.int/search/en/search.jsf	
Guidelines for activity-based assessment: (Conducted for 100 Marks and scaled down to 20 Marks)	

The activity shall be carried out as a structured research survey on an emerging technology and its applications. Students shall adhere to the following guidelines:

- Identification of Research Area

Students shall identify a relevant emerging technology/research area based on the concepts and technologies studied in the previous semester. The selected area should have current research relevance and potential for real-world application.

- Selection of Research Papers

Based on appropriate research keywords, technology domain, and application area of interest, students shall identify and download a minimum of 10 peer-reviewed research papers published in journals indexed in Web of Science and/or Scopus. The selected papers may belong to Q1–Q4 journal quartiles and should preferably represent recent research developments in the identified area.

- Literature Survey and Analysis

Students shall critically review the selected research papers and consolidate the findings. The survey shall address, wherever applicable:

- Research problem and objectives
- Technologies/methodologies employed
- Datasets/tools/frameworks used
- Key findings and performance measures
- Strengths and limitations
- Research gaps identified
- Potential future research directions

Based on the literature survey, students shall prepare an 8–10 page technical paper/report following the IEEE standard paper format. The report shall include appropriate citations and references for all sources.

- Submission and Evaluation

The completed technical paper shall be submitted to the assigned guide/team for evaluation. The work shall be assessed based on prescribed evaluation rubrics, including relevance of the research area, quality and selection of literature, depth of analysis, identification of research gaps, technical presentation, IEEE formatting and referencing.

Evaluation Criteria:

- Research area and application - 10 Marks
- Relevancy of paper selection for review – 10 Marks
- Research problem and objectives, Technologies/methodologies employed, Datasets/tools/frameworks used – 30 Marks
- Key findings, Strengths and limitations, Research gaps identified, Potential future research directions – 30 Marks
- Paper in standard IEEE format – 20 Marks

MACHINE LEARNING			
Semester	II	CIE Marks	50
Course Code	26MCPC516	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:2:2	Exam Hrs	3
Total Hrs	64 (40L + 24P)	Credits	4
Course Description: This course provides a comprehensive foundation in machine learning concepts, algorithms, and practical workflows for solving real-world problems. It covers challenges in machine learning, data exploration and preprocessing, model selection, validation, types of machine learning and performance evaluation. Through practical applications and model evaluation, students develop the ability to build, analyze, fine-tune, and validate machine learning solutions.			
Course Objectives: This course is designed to 1. Provide knowledge on the basics of designing a concept learning system. 2. Impart knowledge on importance of data preprocessing & various preprocessing techniques. 3. Familiarize the working concepts of various supervised & unsupervised machine learning algorithms. 4. Enhance the practical skills in implementing, evaluating, and interpreting machine learning models.			
Module 1: Foundations of Machine Learning			No. of Hrs: 07 + 04
Introduction to Learning System, Designing a Learning System, Perspectives & Issues in Machine Learning, A Concept Learning Task, Concept Learning as Search, Find S Algorithm, Version Space of hypothesis, Candidate Elimination Algorithm, Inductive bias.			
Laboratory components: 1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. 2. Implement and demonstrate the Candidate-Elimination algorithm for a given set of training data examples.			
Self-Study component: Issues in Candidate Elimination Algorithm, Estimating Hypothesis Accuracy, Basics of sampling theory, hypothesis testing			
Textbook 1: 1.1-1.3, 2.1-2.7, 5.1-5.3, 5.5			
Module 2: Data Preprocessing			No. of Hrs: 08 +04
Machine Learning Activities, Data Preprocessing: Types of Data in Machine Learning, Exploring Structure of Data, Data quality issues, Feature engineering: Feature Transformation, Feature Subset Selection, Types of Models: Predictive models, Descriptive models, Model Training & Validation: Steps in Training a Model, Holdout method, K-fold Cross-validation method, Bootstrap sampling, Lazy vs. Eager learner			
Laboratory components: 1. Using a suitable dataset write a Python program to demonstrate following visualizations using Matplotlib: Bar chart, Histogram, Box plot, Scatter plot, Correlation heatmap. Analyze the visualizations to identify data distribution, outliers, relationships, and correlations. Also summarize the observations obtained from the visual analysis. 2. Using a suitable dataset, write a Python program to:			

- Identify and handle missing values and duplicate records.
- Impute missing values using suitable techniques such as mean, median, or mode.
- Detect and handle outliers using the IQR method.
- 3. Using a suitable dataset, write a Python program to:
 - Identify numerical and categorical features.
 - Apply suitable feature transformation
 - Compare the feature distributions before and after transformation.
 - Apply PCA to perform dimensionality reduction

Self-Study component: Practical implementation on SVD, LDA, Measure of feature relevance and redundancy.

Textbook 2: 2, 4, 3.1 – 3.3

Module 3: Supervised Learning

No. of Hrs: 07 + 04

Classification: Classification Model, Steps of Classification Learning, Classification Algorithms: K-nearest Algorithms, Decision Tree **Regression Algorithms:** Simple Linear Regression, Multiple Linear Regression, Issues in Regression Analysis, Improving Accuracy of Regression Models.

Model Evaluation and Performance Improvement: Underfitting, Overfitting, Bias – variance trade-off. Evaluating Performance of classification and regression model. Improve the model performance.

Laboratory components:

1. For a given data set, write a Python program to build a classification model by applying Decision tree and Random forest classification algorithms. Compare & evaluate the models using appropriate classification performance measures
2. For a given data set, write a Python program to build a regression models by applying Linear Regression and Multiple Linear Regression algorithms. Compare & evaluate the model using appropriate regression performance measures.

Self-Study components: Maximum Likelihood Estimation, Practical implementation of Random Forest, Support Vector Machine, Polynomial Regression Model, Logistic Regression.

Textbook 2: 7, 8, 3.4-3.5.2, 3.6

Module 4: Unsupervised Learning

No. of Hrs: 10 + 06

Unsupervised learning and Clustering: Introduction, Application of Unsupervised Learning, Clustering as a machine learning task, **Types of clustering techniques:** Partitioning methods, K-Medoids, Hierarchical clustering, Density-based methods, Evaluating Performance of clustered model.

Pattern Search: Key concepts, Association rule, Apriori algorithm, example of Apriori Algorithm.

Laboratory components

1. Write a Python program to perform unsupervised learning using clustering. Apply K-Medoids and DBSCAN clustering techniques to the dataset and display the resulting clusters.
2. Using a suitable transactional dataset, write a Python program to identify frequent itemsets and generate association rules using the Apriori algorithm. Use appropriate support and confidence values and display the generated rules.

Text Book 2: 9, 3.5.3

Module 5: Basics of Neural Networks

No. of Hrs: 08 + 06

Artificial Neuron and Neural Network Foundations: Biological Neuron, Artificial Neuron, Types of Activation Function. Basic ANN models: McCulloch–Pitts model, Rosenblatt’s perceptron, ADALINE. **Architectures of Neural Network:** Single-layer feed-forward network, Multi-layer feed-forward ANNs, Competitive network, Recurrent network, **Learning Process in ANN:** Number of layers, Direction of signal flow, Number of nodes in layers, Weight of interconnection between neurons, Backpropagation.

Laboratory components:

1. For a given dataset, write a python program to build a classification model with a simple Single Layer ANN architecture.
2. For a given dataset, write a python program to build a classification model with a simple Multi-Layer ANN architecture.

Textbook 2: 10.1 - 10.8

Course Outcomes:

At the end of the course, the student will be able to:

1. Explain the fundamental concepts, principles, and different types of Machine Learning.
2. Use different ML models, data preparation techniques, model validation, and evaluation methods.
3. Apply appropriate Machine Learning technique to solve problems.
4. Evaluate performance of Machine Learning models.

Textbooks:

1. Tom Mitchell, “Machine Learning”, 3rd Edition, McGraw Hill, 2009.
2. Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, Machine Learning, Pearson Education India, 1st Edition, 2018

References:

1. Jason Brownlee, Data Preparation for Machine Learning: Data Cleaning Feature Selection, and Data Transforms in Python, 2020.
2. Andreas C. Müller, Sarah Guido, “Introduction to Machine Learning with Python A Guide for Data Scientists”, 2nd Edition, O’Reilly Publisher, 2020.
3. Aurélien Géron, “Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow”, 2nd Edition, O’Reilly Publisher, 2019.
4. George F. Luger, “Artificial Intelligence: Structures and Strategies for Complex Problem Solving”, 6th Edition, Pearson Education, 2008.
5. Stephen Marsland, “MACHINE LEARNING - An Algorithmic Perspective”, 2nd Edition, CRC Press, 2015.

Web Links:

1. Machine Learning: https://www.w3schools.com/python/python_ml_getting_started.asp
2. Machine Learning Concepts and Algorithms: <https://www.geeksforgeeks.org/machine-learning/>
3. Comprehensive Machine Learning: https://www.tutorialspoint.com/machine_learning/index.htm

Database Management System			
Semester	II	CIE Marks	50
Course Code	26MCPC517	SEE Marks	50
Teaching Hrs/Week (L:T:P:S)	3:0:2:3	Exam Hrs	3
Total Hrs	64 (40L + 24P)	Credits	4
Course Description: This course provides a comprehensive foundation in database management systems, covering database concepts, architecture, users, data models, and database applications. It introduces conceptual database design using the Entity-Relationship model, relational schemas, relational algebra, and integrity constraints. The course develops understanding of functional dependencies and normalization techniques up to Fifth Normal Form. It also covers SQL, including data definition, manipulation, retrieval, constraints, views, and triggers. Finally, the course explores transaction processing, ACID properties, serializability, recoverability, and major concurrency control techniques such as two-phase locking, timestamp ordering, multi-version, and validation-based methods.			
Course Objectives: This course is designed to: 1. Impart knowledge of fundamentals of database management. 2. Provide knowledge of relational data models and relational algebra operations. 3. Familiarize SQL programming. 4. Impart knowledge on normalization and functional dependency. 5. Impart knowledge on concurrency control and transaction management.			
Module 1: Introduction			No. of Hrs: 8L
Characteristics of database approach, Database Users, Advantages, database applications, Disadvantages, Data Models, Schemas, Instances, Three schema architecture, data independence, Database languages, Interfaces, The Database System environment, Centralized and Client/Server architecture of DBMS, Classification of Database Management Systems.			
Textbook 1: 1.1 to 1.8, 2.1 to 2.6			
Module 2: Relational Model and Relational Algebra			No. of Hrs: 8L+6P
Entity Types, Entity Sets, Relationship Types, Relationship Sets, Roles, and Structural Constraints, Weak Entity Types, ER Diagrams, Naming Conventions, and Design Issues Relational Model: Concepts, Constraints and Relational Database Schemas, Update Operations, Transactions, and Dealing with Constraint Violations. Relational Algebra: Unary Relational Operations: SELECT and PROJECT, Relational Algebra Operations from Set Theory, Binary Relational Operations-JOIN and DIVISION, Aggregate Functions, Relational Database Design using ER-to-Relational mapping.			
Laboratory Components: 1. EMPLOYEE (Employee_ID, Employee_Name, Department, Salary, Experience): Each employee is identified by Employee_ID and has the attributes Employee_Name, Department, Salary, and Experience. Write and execute SQL queries for the following requirements: a. Display the names and salaries of employees whose salary is greater than ₹50,000. b. Display the details of employees belonging to the "IT" department, arranged in descending			

order of salary.

2. BOOK (Book_ID, Book_Name, Author, Category, Price): Each book is identified by Book_ID and has the attributes Book_Name, Author, Category, and Price.
Write and execute SQL queries for the following requirements:
 - a. Display the names and prices of books whose price is greater than ₹500.
 - b. Display the details of books belonging to the "Database" category, arranged in ascending order of price.
3. Consider the relation STUDENT (USN, Name, Date_of_Birth, Branch, Mark1, Mark2, Mark3, Total, AVG). Write SQL queries to perform the following operations:
 - a) Calculate and update the Total marks by summing Mark1, Mark2, and Mark3.
 - b) Compute and display the AVG of all students.
 - c) Retrieve the details of students born in a specified year.
 - d) Display the details of students belonging to a specified branch.
 - e) Determine the maximum AVG obtained in each branch.
 - f) Retrieve the details of students whose names begin with the letter 'S'.
 - g) Retrieve the details of students whose names end with 'AR'.
 - h) Delete the record of the student whose USN is 1001.

Self-Study Component: Relational Calculus, Comparison of Relational Algebra and Relational Calculus.

Textbook 1: 3.3 to 3.5.3.7, 5.1 to 5.3, 8.1 to 8.3

Module 3: Normalization

No. of Hrs: 8L+ 6P

Database Design Theory and Normalization: Informal design guidelines for relation schema, Functional Dependencies, First Normal Forms, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.

Laboratory Components:

4. Consider the following database schema:
 BOOK (book_ISBN#, title, publisher, author)
 COURSE (course#, cname, dept)
 BOOK_ADOPTION (course#, sem, book_ISBN#)
 Write SQL queries to perform the following operations:
 - a) Display the details of books adopted for each course.
 - b) List the books published by a specified publisher.
 - c) Retrieve the courses that have adopted books from a particular publisher.
 - d) Find the number of books adopted for each course.
 - e) Display publishers who have published more than one adopted book.
 - f) Display the titles and authors of all books adopted for a specified course.
 - g) List all courses that have not adopted any books.
 - h) Find the course that has adopted the maximum number of books.
 - i) Display the details of books that have been adopted by more than one course.
 - j) Find the total number of distinct books adopted by each department.

5. i. Design an ER Diagram for the following scenario.

A University organizes an annual Inter-College Sports Championship in which multiple colleges participate.

COLLEGE (College_ID, College_Name, City, Coach_Name, Captain_ID): Each college is uniquely identified by College_ID and has a College_Name, City, and Coach_Name. One player from each college is designated as the Team Captain.

PLAYER (Player_ID, Player_Name, Age, Gender, Phone_Number, College_ID): Each player is uniquely identified by Player_ID and has Player_Name, Age, Gender, and Phone_Number. Each player belongs to only one college, while a college can have many players.

EVENT (Event_ID, Event_Name, Event_Type): The championship consists of several sports events. Each event is uniquely identified by Event_ID and has an Event_Name and Event_Type.

VENUE (Venue_ID, Venue_Name, City, Capacity): Events are conducted at different venues. Each venue is uniquely identified by Venue_ID and has a Venue_Name, City, and Capacity.

MATCH (Match_ID, Event_ID, Venue_ID, College1_ID, College2_ID, Scheduled_Date, Scheduled_Time, Winning_College_ID, Best_Player_ID): Each match is uniquely identified by Match_ID and is played between two colleges for a particular event at a specific venue. The match records the scheduled date and time, participating colleges, winning college, and Best Player of the Match.

Captain: Each college has one Team Captain, who must be one of the players belonging to that college.

Best Player: Each match has one Best Player, selected from the players of the two participating colleges. A player can receive the Best Player award in multiple matches.

ii. Convert the ER Diagram into a Relational Model

iii. Write and Execute SQL Queries for the Following Requirements

- Display the name, age, and college name of the youngest player participating in the championship.
- Display the details of the venue where the maximum number of matches were conducted.
- Display the details of players who are not captains but have received the Best Player of the Match award in at least two matches.
- Display the details of the college that has won the maximum number of matches.
- List the college names whose all winning matches were played at the same venue.
- Display the details of all matches conducted for a specified sports event, including the participating colleges, venue, scheduled date, and scheduled time.
- Find the number of matches conducted for each sports event.
- Display the details of venues whose capacity is greater than the average capacity of all venues.
- List the details of colleges that participated in the championship but did not win any match.

6. i. Design an ER Diagram for the following scenario.

A country has many National Parks.

NATIONALPARK (Park_ID, Park_Name, State, Established_Year, Area): Each national park is uniquely identified by Park_ID and has the attributes Park_Name, State, Established_Year,

and Area. A state can have multiple national parks.

VISITOR (Visitor_ID, Visitor_Name, Age, Country, Contact_Numbers): Each visitor is uniquely identified by Visitor_ID and has the attributes Visitor_Name, Age, and Country. A visitor may have multiple Contact_Numbers.

VISIT (Visitor_ID, Park_ID, Visit_Date, Duration_of_Stay): A visitor can visit many national parks, and a national park can be visited by many visitors. For each visit, Visit_Date and Duration_of_Stay are recorded. A visitor may visit the same national park multiple times on different dates.

Based on the above scenario, design the ER Diagram showing all entities, attributes, primary keys, foreign keys, relationships, and cardinalities.

- ii. Convert the ER Diagram into a Relational Model and normalize the relations into a suitable Normal Form.
- iii. Write and execute SQL queries for the following requirements:
 - a) List the state that has the maximum number of national parks.
 - b) Display the details of the national park that has been visited by the maximum number of visitors.
 - c) List the details of visitors who have visited all national parks located in the state "KARNATAKA".
 - d) Display the details of visitors who have visited at least one national park in every state recorded in the database.
 - e) List the details of national parks that have been visited by visitors from all countries recorded in the database.
 - f) Display the names of visitors along with the total number of national parks they have visited.
 - g) Find the national park with the longest average duration of stay among visitors.
 - h) List the visitors who have visited the same national park more than once on different dates.
 - i) Display the state-wise number of visitors who have visited national parks.
 - j) Find the total number of visits and average duration of stay for each national park.
 - k) List the national parks that have never been visited by any visitor.
 - l) Display the details of visitors whose age is greater than the average age of all visitors.

Self-Study Component: Inference Rules, Multivalued Dependencies, Denormalization.

Textbook 1: 14.1, 14.3 to 14.7.

Module 4: Structured Query Language

No. of Hrs: 8L+6P

SQL data definition, data types, constraints, Retrieval queries in SQL, DDL, DML and DCL statements in SQL, Complex SQL retrieval queries, Specifying constraints as assertions, Triggers and Views.

Laboratory Components:

7. i. Design an ER Diagram for the following scenario.

An airline operates several flights and serves many passengers.

FLIGHT (Flight_ID, Flight_Name, Source, Destination, Departure_Time, Arrival_Time, Fare, Pilot_ID): Each flight is uniquely identified by Flight_ID and has the attributes Flight_Name, Source, Destination, Departure_Time, Arrival_Time, and Fare. Each flight is operated by one pilot, while a pilot may operate multiple flights.

PASSENGER (Passenger_ID, Passenger_Name, Age, Gender, Phone_Number, City): Each passenger is uniquely identified by Passenger_ID and has the attributes Passenger_Name, Age, Gender, Phone_Number, and City. A passenger can book multiple flights.

PILOT (Pilot_ID, Pilot_Name, License_Number, Experience, Phone_Number): Each pilot is uniquely identified by Pilot_ID and has the attributes Pilot_Name, License_Number, Experience, and Phone_Number. A pilot may operate multiple flights, but each flight is operated by only one pilot.

BOOKING (Booking_ID, Passenger_ID, Flight_ID, Booking_Date, Seat_Number, Travel_Date, Booking_Status): Each booking is uniquely identified by Booking_ID and records the booking made by a passenger for a particular flight. It has the attributes Passenger_ID, Flight_ID, Booking_Date, Seat_Number, Travel_Date, and Booking_Status. A passenger can make multiple bookings, and a flight can have many bookings.

- ii. Based on the above scenario, design the ER Diagram showing all entities, attributes, primary keys, foreign keys, relationships, and cardinalities. Convert the ER Diagram into a Relational Model and normalize the relations into a suitable Normal Form.
- iii. Write and execute SQL queries for the following requirements:
 - a) Display the details of the flight that has the maximum number of bookings.
 - b) List the details of passengers who have travelled to more than one destination.
 - c) Display the details of the pilot who operates the maximum number of flights.
 - d) List the passengers who have booked flights to all destinations available in the database.
 - e) Display the details of flights that have no bookings.
 - f) Display the total number of bookings made for each flight, along with the Flight_Name.
 - g) List the passengers who have made more than one booking.
 - h) Find the total revenue generated by each flight based on the fare of the booked tickets.
 - i) Display the details of passengers who have never booked any flight.
 - j) List the flights operated by pilots having more than 10 years of experience.
 - k) Display the Source, Destination, and number of passengers booked for each flight.
 - l) Find the passenger who has made the maximum number of bookings.

8. i. Design an ER Diagram for the following scenario.

A country has several hospitals.

HOSPITAL (Hospital_ID, Hospital_Name, City, State, Capacity, Doctor_Count): Each hospital is uniquely identified by Hospital_ID and has the attributes Hospital_Name, City, State, Capacity, and Doctor_Count. A hospital can employ many doctors.

DOCTOR (Doctor_ID, Doctor_Name, Specialization, Age, Phone_Number, Experience, Hospital_ID): Each doctor is uniquely identified by Doctor_ID and has the attributes Doctor_Name, Specialization, Age, Phone_Number, and Experience. Each doctor works in only one hospital, while a hospital can have many doctors.

PATIENT (Patient_ID, Patient_Name, Age, Gender, Address, Contact_Number): Each patient is uniquely identified by Patient_ID and has the attributes Patient_Name, Age, Gender, Address, and Contact_Number. A patient may consult multiple doctors, and a doctor may consult multiple patients.

CONSULTATION (Consultation_ID, Doctor_ID, Patient_ID, Consultation_Date, Diagnosis, Fee): Each consultation is uniquely identified by Consultation_ID and records the Doctor_ID, Patient_ID, Consultation_Date, Diagnosis, and Fee. A doctor can have many consultations, and a patient can have many consultations.

- ii. Based on the above scenario, design the ER Diagram showing all entities, attributes, primary keys, foreign keys, relationships, and cardinalities. Convert the ER Diagram into a Relational Model and normalize the relations into a suitable Normal Form.
- iii. Write and execute SQL queries for the following requirements:
 - a) List the details of doctors who have consulted patients from more than one city.
 - b) Display the state that has the maximum number of hospitals.
 - c) Create a TRIGGER to automatically update the Doctor_Count of the respective hospital whenever a new doctor is inserted into the DOCTOR table.
 - d) List the hospital names along with the number of doctors working in each hospital.
 - e) Display the details of patients who have consulted more than one doctor.
 - f) Find the doctor who has conducted the maximum number of consultations.
 - g) Display the details of all consultations, including Patient_Name, Doctor_Name, Hospital_Name, Consultation_Date, Diagnosis, and Fee.
 - h) Find the total consultation fee collected by each hospital.

Self-Study Component: Database Stored Procedures, SQL Window Functions (ROW_NUMBER(), RANK(), DENSE_RANK(), OVER Clause).

Textbook 1: 6.1 to 6.5, 7.1 to 7.3.

Module 5: Transaction Management and Concurrency Control

No. of Hrs: 8L+6P

Transaction Processing, ACID Properties, characterizing schedules based on Recoverability and Serializability, Transaction support in SQL. Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multi version Concurrency control techniques, Validation Techniques and Snapshot Isolation Concurrency Control

Laboratory Components:

9. Stored Procedure – Bank Account Management System

You are managing a Bank Account Management System.

BANK_ACCOUNT (Account_ID, Customer_Name, Account_Type, Balance,

Minimum_Balance): Each account is identified by Account_ID and contains the customer name, account type, balance, and minimum required balance.

Withdrawal: Create a stored procedure that accepts Account_ID and Withdrawal_Amount as input parameters. The procedure should check whether sufficient balance is available while maintaining the Minimum_Balance. If permitted, deduct the withdrawal amount and display the updated balance; otherwise, display an appropriate message indicating that the withdrawal cannot be processed.

10. Trigger – Product Inventory Management System

You are managing the backend of a Product Inventory Management System for a retail company.

PRODUCT (Product_ID, Product_Name, Category, Unit_Price, Quantity_Available): Each product is identified by Product_ID and contains the product name, category, unit price, and available quantity.

Trigger: Create triggers to automatically track changes whenever a new product is inserted, an existing product's price is updated, or a product record is deleted. For a price update, the trigger should calculate and display the difference between the old price and the new price.

Textbook 1: 20.1, 20.3, 20.4, 20.6, 21.1-21.4

Course Outcomes:

At the end of the course, the student will be able to:

1. Describe the concepts of relational databases, transactions, and concurrency control techniques.
2. Apply relational data modelling techniques in database design.
3. Apply normalization techniques to eliminate data redundancy and improve data integrity.
4. Write SQL queries using DDL, DML, and DCL statements.

Textbooks

1. Elmasri, R., & Navathe, S. B., “Fundamentals of Database Systems”, 7th Edition, Pearson, 2017.

Reference Books

1. Silberschatz, A., Korth, H. F., & Sudarshan, S., “Database System Concepts”, 6th Edition, Tata McGraw Hill Education Private Limited, 2011.
2. Ramakrishnan, R., & Gehrke, J., “Database Management Systems”, 3rd Edition, McGraw Hill, 2014.

Web links:

1. Database Management System: https://onlinecourses.nptel.ac.in/noc22_cs91/preview
2. Introduction to Databases: <https://archive.nptel.ac.in/courses/106/106/106106220/>
3. Database Management Essentials: <https://www.coursera.org/learn/database-management>