1st Semester Syllabus

Semester: I

Course Name: **PROBLEM SOLVING USING C**

Course Code	MMC101	CIE Marks	50
Teaching Hours/Week (L:P: T)	3:2:0	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 20 hours Lab	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

1. Apply conditional and branching statements to develop C programs that solve decision-making problems.
2. Construct C programs using arrays and strings to efficiently store, process, and manipulate data.
3. Organize modular applications by designing and implementing reusable functions in C.
4. Integrate pointers and structures to develop flexible and dynamic C applications.
5. Implement input/output operations and file handling techniques to manage data persistence in C programs.

Module – 1

BASICS OF C PROGRAMMING

Introduction to programming paradigms, Applications of C Language, Structure of C program, Data Types, Constants, Keywords, Operators, Precedence and Associativity, Expressions, Input/Output statements, Decision making statements, Switch statement, Looping statements Preprocessor directives, Compilation process **8 Hours**

Module – 2

ARRAYS AND STRINGS

Introduction to Arrays: Declaration, Initialization One dimensional array, Two dimensional arrays - String operations: length, compare, concatenate, copy, bubble sort, linear and binary search.

8 Hours

Module – 3

FUNCTIONS AND POINTERS

Modular programming - Function prototype, function definition, function call, Built-in functions, Recursion, Pointers, Pointer arithmetic, Arrays and pointers, Pass by value.

8 Hours

Module – 4

STRUCTURES AND UNION

Structure - Nested structures, Pointer and Structures Array of structures, Self-referential structures Dynamic memory allocation, typedef, Union - Storage classes and Visibility

8 Hours

Module – 5

FILE PROCESSING

Files Types of file processing: Sequential access, Random access, Sequential access file - Random access file, Command line arguments. **8 Hours**

Lab Component:

1. A team of two or four students must develop a course project using C programming constructs learnt in the theory which should include the following concepts
   - Decision making statements
   - Looping statements
   - Arrays & Strings
   - Functions & Pointers
   - Structures & Unions
   - File handling
2. The team must submit a brief project report (20-25 pages) that must include the following
   a. Introduction
   b. Requirement Analysis
   c. Software Requirement Specification
   d. Design and Implementation
   e. Validation
3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It was recommended that students to do prior art search as part of literature survey before submitting the synopsis for the Course-project projects.
4. Rubrics may be used to evaluate the Course-Project

Course Outcomes:

The students should be able to:	
CO1	Develop simple applications in C by utilizing basic programming constructs
CO2	Design and implement applications in C by applying arrays and strings for data organization and manipulation
CO3	Design and implement modular applications in C using functions and pointers to enhance code reusability and efficiency
CO4	Develop structured applications in C by effectively integrating structures for complex data representation
CO5	Create applications using sequential and random-access file processing techniques to manage persistent data storage

Reference Books:

1. "The C programming language", Brian W. Kernighan and Dennis M. Ritchie, Pearson, 2nd Edition, 2015.
2. "Programming in ANSI C", E. Balaguruswamy, Tata McGraw-Hill., 9th Edition, 2024.
3. "Computer fundamentals and programming in C", Reema Thareja, Oxford University, 3rd Edition, 2023.

Semester: I

Course Name: **DISCRETE MATHEMATICS, GRAPH THEORY, PROBABILITY AND STATISTICS**

Course Code	MMC102	CIE Marks	50
Teaching Hours/Week (L:P: T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Interpret, identify, and solve the language associated with logical structure, sets, relations and functions, modular arithmetic
2. To develop the knowledge of matrices and linear algebra in a comprehensive manner
3. Understand an intense foundational introduction to fundamental concepts in discrete Mathematics
4. Applying discrete and continuous probability distributions in analyzing the probability models arising in engineering field.
5. Students will achieve command of the fundamental definitions and concepts of graph theory

Module – 1

SETS, RELATIONS AND FUNCTIONS

Cartesian product of sets. Relations, Properties of relations, Zero-one matrices and directed graphs, Hasse diagram, Equivalence relations and partitions. Functions-types of functions: composition function and Inverse function, Permutation of functions.

8 Hours

Module – 2

LINEAR ALGEBRA

Linear Transformation, Elementary row transformation, Rank of a matrix, Consistency and Solution of system of linear equations, Solution of system of non-homogeneous equations by Gauss elimination method and Gauss Jordon method, Eigen value and the corresponding Eigenvector by power method.

8 Hours

Module – 3

MATHEMATICAL LOGIC

Propositions- Truth Value, Conjunction, Disjunction, Negation, Implication, Converse, Inverse, Contra positive, Bi conditional, Order of Precedence, Tautology, Contradiction,

Logical Equivalences, Quantifiers: Predicates, De Morgan's laws, Arguments: valid and invalid arguments, rules of inference

8 Hours

Module – 4

INTRODUCTION TO GRAPHS

Application of graphs finite, infinite and bipartite graphs Incidence and Degree Isolated vertex, pendant vertex and Null graph, Paths and circuits Isomorphism, subgraphs, walks, paths and circuits, connected graphs, disconnected graphs and components.

8 Hours

Module – 5

PROBABILITY DISTRIBUTIONS

Random variables- discrete and continuous, probability mass function, probability density function, Cumulative density function, Binomial distribution, Poisson distribution, Exponential distribution and Normal distribution. (Only examples)

8 Hours

Course Outcomes:

The students should be able to:	
CO1	Analyze the concepts of relations to various fields of Engineering
CO2	Solve the system of equations using matrix theory and compute Eigen values and Eigen vectors required for matrix diagonalization process
CO3	Apply the concepts of logic for effective computation and relating problems in the Engineering domain
CO4	Apply the basic concepts of graph theory and judge the planar graphs
CO5	Applying discrete and continuous probability distributions in analyzing the probability models arising in engineering field

Reference Books:

1. "Higher Engineering Mathematics", B.S. Grewal, Khanna Publishers, 44th Edition, 2017
2. "Graph theory with applications to engineering and computer science", Narsingh Deo, Dover publishers, 1st Edition 2016
3. "Foundation of Discrete Mathematics", K D Joshi, New Age Publishers, Ltd, 10th Edition, 2014

Semester: I

Course Name: **DATABASE MANAGEMENT SYSTEMS**

Course Code	MMC103	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Analyze the fundamental concepts and diverse applications of database systems to understand their role in real-world scenarios.
2. Evaluate the key challenges and considerations in designing and implementing efficient and effective database systems.
3. Explain the principles of the relational data model, entity-relationship modeling, relational database design, relational algebra, and the SQL database language.
4. Design and construct a simple database system while demonstrating proficiency in modeling, designing, and implementing database management systems (DBMS).
5. Analyze and implement concepts of normalization, transaction processing, concurrency control, and database recovery protocols to ensure reliable database operations

Module – 1

INTRODUCTION

Data, Database, DBMS, Data models, Characteristics of database approach, Actors on the scene, Workers behind the scene, Advantages of using DBMS approach, Three schema architecture of DBMS, Database languages and interfaces, Classification of DBMS.

8 Hours

Module – 2

DATA MODELS

Introduction to the Relational Model Structure Database Schema, Keys Schema Diagrams E/R Model Conceptual data modelling**:** design, Entities, Entity types, entity sets, attributes, keys, relationships, relationship types, E/R diagram notation, examples, Database design other models, Relational Query Languages.

RELATIONAL ALGEBRA

Selection and projection set operations renaming Joins Division Examples of Algebra, overviews Relational calculus Tuple Relational Calculus (TRC) Domain Relational Calculus (DRC).

8 Hours

Module – 3

STRUCTURED QUERY LANGUAGE(SQL)

Basic structure of SQL queries, data types, creating a database, create a table, drop the database, drop table, select table, insert a record, update record, delete a record, order by, group by, triggers, Set Operations, Aggregate Functions, Nested Sub queries, Views, Procedures, Cursors, PL/SQL.

8 Hours

Module – 4

DATABASE PROGRAMMING

Introduction, Nonloss decomposition and functional dependencies, First, Second, and third normal forms dependency preservation, Boyce/Codd normal form.

Higher Normal Forms Introduction, Multivalued dependencies and Fourth normal form, Join dependencies and Fifth normal form.

8 Hours

Module – 5

TRANSACTION MANAGEMENT

Transaction Processing: Introduction to transaction processing, transaction and system concepts, desirable properties of transactions, transaction support in SQL.

Concurrency control techniques: two-phase locking techniques, concurrency control based on timestamp ordering, multi-version concurrency control techniques, validation concurrency control techniques, Case Study on Database Architecture.

8 Hours

Course Outcomes:

The students should be able to:	
CO1	Summarize the fundamental concepts of database objects and identify appropriate data models to solve relevant problems
CO2	Design entity-relationship diagrams and convert them into relational database schemas, while formulating SQL queries for data management and retrieval
CO3	Use Structured Query Language (SQL) to perform database manipulation tasks such as data insertion, updating, deletion, and querying
CO4	Implement normalization techniques based on database design theory to optimize database structures for various applications
CO5	Analyze and implement transaction processing, concurrency control mechanisms, and database recovery protocols to ensure data consistency and reliability in databases

Reference Books:

1. "Fundamentals of Database Systems", Elmasri and Navathe, Pearson Education, 7th Edition, 2017.
2. "Database Management Systems", Raghu Ramakrishnan, Johannes Gehrke, TATA McGraw Hill, 3rd Edition, 2002
3. "Expert Oracle Database Architecture", Thomas Kyte and Darl Kuhn, Apress, 3rd Edition, 2014.

Semester: I

Course Name: **OPERATING SYSTEM**

Course Code	MMC104	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Understand the fundamental concepts and functionalities of operating systems.
2. Explain the mechanisms used by operating systems to handle and manage processes efficiently.
3. Gain knowledge of techniques for preventing and detecting deadlocks in operating systems.
4. Study and analyze various mechanisms involved in memory management techniques, including allocation and paging.
5. Analyze the functions and techniques used in file system management for data storage and retrieval

Module – 1

INTRODUCTION

Operating Systems, System structure, what operating systems do, Operating System Operations, Computing Environments, Operating System Services, System Calls, Types of System Calls, System Programs, Operating System Structure, System Boot.

8 Hours

Module – 2

PROCESS MANAGEMENT

Process Concept: Process states, PCB, Process creation and termination, Process Scheduling, Threads concepts, thread scheduling and Interprocess Communication

Process scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Synchronization Background, The Critical Section Problem, Mutex Locks, Semaphores

Classic Problems of Synchronization: Readers-Writers Problem, Dining Philosophers Problem using Semaphores. Case study using UNIX OS.

8 Hours

Module – 3

DEADLOCKS

System model, Deadlock characterization, Methods for handling deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from Deadlock

8 Hours

Module – 4

MEMORY MANAGEMENT

Basic Hardware, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory Management Background, Demand Paging, Page Replacement, Case study using UNIX OS.

8 Hours

Module – 5

FILE SYSTEM MANAGEMENT File concept, Access methods, Directory overview Implementing File System Allocation methods, Free Space Management, Case study using UNIX OS. **8 Hours**

Course Outcomes:

The students should be able to:	
CO1	Describe the core elements and various functionalities of an operating system
CO2	Apply process management techniques and demonstrate methods for achieving process synchronization
CO3	Understand and explain the concepts and techniques related to deadlock prevention, avoidance, and detection
CO4	Analyze different memory management strategies, including virtual memory, to evaluate their efficiency and effectiveness
CO5	Understand and explain the principles and functionalities of file system management in operating system

Reference Books:

1. "Operating Systems Principles", Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Wiley – India, 10th Edition, 2021.
2. "Operating Systems A Concept Based Approach", D M Dhamdhere, Tata McGraw Hill, 3rd Edition, 2017.
3. "Operating Systems", Harvey M Deital, Addison Wesley, 3rd Edition, 1990

Semester: I

Course Name: **WEB TECHNOLOGIES**

Course Code	MMC105	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Understand the fundamentals of the Internet and the foundational principles of web design.
2. Apply knowledge of HTML and Cascading Style Sheets (CSS) to construct basic websites.
3. Analyze and build dynamic web pages by implementing JavaScript objects and utilizing various event-handling mechanisms for validation.
4. Create interactive web pages by integrating and applying jQuery effects.
5. Develop simple web applications using PHP for enhanced functionality and user interaction

Module – 1

WEB BROWSERS, WEB SERVERS, MIME, URL, HTTP

Introduction to HTML tags, Basic syntax and structure, Text Markups - paragraphs, line breaks, white spaces, headings, block quotations, font style and size, character entity. Images, Lists, Tables, Links, Frames, meta tag, span and div tags.

8 Hours

Module – 2

CASCADING STYLE SHEETS (CSS)

Introduction to CSS, Levels of CSS, Selectors, Font, Color and Text Properties, Box Model, Introduction to JavaScript, JavaScript variables, operators, Conditional and loop statements in JavaScript, Functions and Arrays in JavaScript, Validations using JavaScript, Screen output and Keyboard input.

8 Hours

Module – 3

DOM AND EVENT HANDLING

Document Object Model in JavaScript, Handling Strings and working with Window object, Handling events - Mouse events, Keyboard events, form events and window/document events, using addEventListener() method.

8 Hours

Module – 4

JQUERY

Syntax, selectors, events, JQuery Effects- jQuery hide/show, fade, slide, animate, stop, delay, JQuery HTML-get, set, add and remove, JQuery CSS-jQuery CSS, before, after, append.

8 Hours

Module – 5

PHP Overview of PHP, General Syntactic Characteristics, Primitives, Operations and Expressions, Output, Control statements, Arrays, Functions, Form Handling. **8 Hours**

Course Outcomes:

The students should be able to:	
CO1	Describe the components of a web page and create simple web pages while applying appropriate formatting techniques
CO2	Apply CSS features to style web pages according to application requirements and use JavaScript for validation
CO3	Identify the Document Object Model (DOM) and develop dynamic web pages by implementing JavaScript event-handling mechanisms
CO4	Construct interactive web pages by integrating jQuery effects and manipulating HTML/CSS using jQuery
CO5	Develop simple web pages using PHP and implement server-side scripting to dynamically generate web content

Reference Books:

1. "Programming the World Wide Web", Robert W. Sebesta, Pearson Publication, 8th Edition, 2020
2. "Beginning JQuery", Jack Franklin, APRESS, 2nd Edition. 2017
3. "Web Programming", Chris Bates, Wiley Publications, 3rd Edition, 2007

Semester: I

Course Name: **DBMS AND WEB TECHNOLOGIES LABORATORY**

Course Code	MMCL106	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:4:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	02	Exam Hours	03

Course Objectives:

1. Understand Fundamental Database Concepts, their applications, data models, schemas, and instances.
2. Design and Implement Database Objects such as tables, constraints, procedures, and sequences.
3. Apply database normalization techniques and its principles to design efficient databases
4. Construct basic websites using HTML, jQuery, and Cascading Style Sheets (CSS) for structure, interactivity, and design.
5. Create dynamic web pages with client-side validation using JavaScript objects, applying various event-handling mechanisms.

LAB COMPONENT

1. A team of two or four students must develop a course project using DBMS and Web Technology constructs learnt in the theory which should include the following concepts
   - Design an ER Diagram, Schema, Creating, Altering, Dropping tables with Constraints
   - Create tables, alter table, Drop tables
   - Aggregate functions & Grouping clauses
   - Using character functions, number functions, date functions, EQUI-JOINS, correlated sub-queries
   - Grouping, Joins, Procedures
   - PL/SQL, Triggers, Cursors
   - HTML tags
   - Javascript script operations
   - Design, test and validate an HTML document that contain check boxes
   - CSS
   - JQUERY
   - PHP

2. The team must submit a brief project report (20-25 pages) that must include the following
   i. Introduction
   ii. Requirement Analysis
   iii. Software Requirement Specification
   iv. Design and Implementation
   v. Validation

3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It was recommended that students to do prior art search as part of literature survey before submitting the synopsis for the Course-project projects.

4. Rubrics may be used to evaluate the Course-Project

Course Outcomes:

The students should be able to:	
CO1	Apply database concepts and technologies to create relationships by specifying primary and foreign keys
CO2	Design an Entity-Relationship Diagram (ERD) and implement a corresponding database schema for a given problem
CO3	Apply normalization techniques to refine and improve the database design for the given problem
CO4	Develop web pages using HTML, jQuery, JavaScript, and CSS, with diverse layouts suited to application requirements
CO5	Implement dynamic web applications using PHP for backend functionality and real-time interactivity.

Semester: I

Course Name: **RESEARCH METHODOLOGY AND IPR (NCMC COURSE)**

Course Code	MRMI107	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40	Total Marks	100
Credits	0	Exam Hours	03

Course Objectives:

1. Demonstrate the ability to choose methods appropriate to research aims and objectives
2. Understand the limitations of particular research methods
3. Analyze experimental design with the samplings
4. Develop skills in qualitative and quantitative data analysis and presentation
5. Understand Intellectual Property and its importance

Module – 1

RESEARCH METHODOLOGY

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, and Problems Encountered by Researchers in India

8 Hours

Module - 2

DEFINING THE RESEARCH PROBLEM

Research Problem, Selecting the Problem, Necessity of Defining the Problem, Technique Involved in Defining a Problem, An Illustration. Reviewing the literature: Place of the literature review in research, Bringing clarity and focus to your research problem, Improving research methodology, Broadening knowledge base in research area, Enabling contextual findings, How to review the literature, searching the existing literature, reviewing the selected literature, Developing a theoretical framework, Developing a conceptual framework, Writing about the literature reviewed.

8 Hours

Module – 3

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, Important Experimental Designs. Design of Sample Surveys: Introduction, Sample Design, Sampling and Non-sampling Errors, Sample Survey versus Census Survey, Types of Sampling Designs **8 Hours**

Module – 4

DATA COLLECTION

Experimental and Surveys, Collection of Primary Data, Collection of Secondary Data, Selection of Appropriate Method for Data Collection, Case Study Method. Interpretation and Report Writing: Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation, Significance of Report Writing, Different Steps in Writing Report, Layout. Types of Reports, Oral Presentation, Mechanics of Writing a Research Report, Precautions for Writing Research Reports.

8 Hours

Module – 5

INTELLECTUAL PROPERTY (IP) ACTS
Introduction to IP: Introduction to Intellectual Property (IP), different types of IPs and its importance in the present scenario, Patent Acts: Indian patent acts 1970.Design Act: Industrial Design act 2000. Copy right acts: Copyright Act 1957. Trade Mark Act, 1999 **8 Hours**

Course Outcomes:

The students should be able to:	
CO1	Identify the suitable research methods and articulate the research steps in a proper sequence for the given problem.
CO2	Carry out literature survey, define the problem statement and suggest suitable solution for the given problem and present in the format of the research paper (IEEE).
CO3	Analyze the problem and conduct experimental design with the samplings
CO4	Perform the data collection from various sources segregate the primary and secondary data
CO5	Apply some concepts/section of Copy Right Act /Patent Act /Cyber Law/ Trademark to the given case and develop –conclusions

Reference Books:

1. "Research Methodology: Methods and Techniques", C.R. Kothari & Gaurav Garg, New Age International, 5th Edition, 2023.
2. "Research Methodology a step-by- step guide for beginners", Ranjit Kumar, SAGE Publications Ltd, 4th Edition, 2024.
3. "Intelectual property", Deborah E. Bouchoux, Cengage learning,4th Edition, 2013

2nd Semester Syllabus

Semester: II

Course Name: **DATA ANALYTICS USING PYTHON**

Course Code	MMC201	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:2:0	SEE Marks	50
Total Hours of Pedagogy	40 hours Theory + 20 hours lab	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

1. Understand foundational concepts in data analytics, principles, techniques, and their applications
2. Explore and apply basic programming constructs in Python to solve computational problems
3. Implement and utilize Python data structures, such as lists, dictionaries, and sets, to address real-world challenges
4. Create and implement solutions leveraging Python libraries NumPy and Pandas for data manipulation and analysis
5. Perform data preprocessing, create insightful visualizations, and evaluate models to derive meaningful insights from data

Module – 1

INTRODUCTION

What is data, Big data analytics, Types of analytics, Big data analytics framework, Descriptive analytics, Predictive analytics, Univariate data analysis and Visualization

8 Hours

Module – 2

PYTHON BASIC CONCEPTS AND PROGRAMMING

Variables, Operators, Keywords, Statements and Expressions, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, The type( ) Function and Is Operator, Control Flow Statements, The if Decision Control Flow Statement, The if…else Decision Control Flow Statement, The if…elif…else Decision Control Statement, Nested if Statement, The while Loop, The for Loop, The continue and break Statements, Built-In Functions, Commonly Used Modules, Function Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of Variables, Default Arguments

8 Hours

Module – 3

PYTHON COLLECTION OBJECTS

Classes Strings- Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings, Lists- Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, Tuples and Dictionaries, Files: reading and writing files

8 Hours

Module – 4

NUMPY AND PANDAS

Numpy: Understanding datatypes in python, Basics of NumPy arrays, Computation on NumPy arrays: Universal functions, Pandas-Introducing to Pandas data structures, Essential functionality, Summarizing and Computing descriptive statistics, Handling missing data.

8 Hours

Module – 5

VISUALIZATION WITH MATPLOTLIB AND SEABORN

General Matplotlib tips, Simple line plots, Simple scatter plots, Visualizing errors, Density and contour plots, Histograms, Binning, and density, Customizing plot legends and Colorbars, Customizing matplotlib, Visualization with Seaborn.

8 Hours

Lab Component:

1. A team of two or four students must develop a course project using C programming constructs learnt in the theory which should include the following concepts
   - Definite and Indefinite loops
   - Fruitful and non-fruitful functions
   - String operations
   - list, tuple, dictionary, files
   - NumPy
   - Pandas
   - Matplotlib
   - Seaborn

2. The team must submit a brief project report (20-25 pages) that must include the following
   i. Introduction
   ii. Requirement Analysis
   iii. Software Requirement Specification
   iv. Design and Implementation
   v. Validation
3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It was recommended that students to do prior art search as part of literature survey before submitting the synopsis for the Course-project projects.

4. Rubrics may be used to evaluate the Course-Project

Course Outcomes:

The students should be able to:	
CO1	Demonstrate a clear understanding of foundational concepts, techniques, and applications in data analytics
CO2	Understand and utilize Python programming constructs to address basic computational challenges
CO3	Implement Python data structures, such as lists, tuples, dictionaries, and sets, to solve practical problems in real-world contexts
CO4	Create and implement applications that leverage Python libraries like NumPy and Pandas for data manipulation and analysis
CO5	Analyze and assess data preprocessing methods, visualization techniques, and model evaluation processes to extract actionable insights

Reference Books:

1. "Think Python: How to Think Like a Computer Scientist", Allen B. Downey, Shroff/O'Reilly, 2nd Edition, 2021
2. "Python Data Science Handbook: Essential tools for working with data", Jake Vander plas, O'Reilly Media, Inc., 2nd Edition, 2022
3. "Programming Python", Mark Lutz, O'Reilly Media, 5th Edition, 2017

Semester: II

Course Name: **OBJECT ORIENTED PROGRAMMING USING JAVA**

Course Code	MMC202	CIE Marks	50
Teaching Hours/Week (L:P: T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50 hours	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

1. Familiarize the principles of object-oriented programming
2. Illustrate Inheritance and Generics concepts
3. Implement Packages, Interfaces and Event Handling to solve real world problems.
4. Familiarize the concepts of Multithreading and Input/Output streams
5. Demonstrate UI Programming and Advance Java Programming

Module – 1

INTRODUCTION

The Byte code, Features of Java, Object- Oriented Programming principles, Structure of a Java program, Data Types and Variables, Type conversion and casting, Arrays

Classes: Fundamentals, Declaring Objects, Assigning Object Reference Variables, Methods, Constructors, this Keyword, Garbage Collection.

Methods and Classes: Overloading Methods, Using Objects as Parameters, Argument Passing, Returning Objects, Access Control, static, final, Command-Line Arguments

10 Hours

Module – 2

INHERITANCE

Basic concepts, Member Access and Inheritance, Practical Example Inheritance types, super, constructors, Method Overriding, Dynamic Method Dispatch, Abstract Classes, final with inheritance.

GENERICS

About Generics, A simple Generic Example, General class with Two Type Parameters, General form of generic class. Introduction to Lambda expressions, Block Lambda Expressions, Generic functional Interfaces.

10 Hours

Module – 3

PACKAGES AND INTERFACES

Packages, Packages and member access, importing packages, Interfaces, Default interface methods, Use static methods in an interface, Private Interface methods.

EXCEPTION HANDLING

Fundamentals, Exception types, uncaught exceptions, try and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's build-in exceptions, User- defined exceptions.

10 Hours

Module – 4

MULTITHREADED PROGRAMMING

Java thread model, main thread, creating thread, creating multiple threads, isalive( ) and Join( ), thread priorities, synchronization

INPUT/OUTPUT

Exploring java.io - The I/O Classes and Interfaces, The Byte Streams

10 Hours

Module – 5

UI PROGRAMMING

Swings, UI components, event handling, Two Event Handling Mechanisms, The Delegation Event Model, Events Event Sources, Event Listeners, Event Classes- The MouseEventClass, KeyEventClass.

Advance Java Programming: JDBC, Network Programming, Servlets, JSP.

10 Hours

Course Outcomes:

The students should be able to:	
CO1	Apply object-oriented programming principles to implement simple Java programs
CO2	Demonstrate the Inheritance and Generics
CO3	Apply packages, interfaces concepts in solving real world problems
CO4	Design and develop multithreaded programming
CO5	Implement advance Java programming tools to solve real world scenarios

Reference Books:

1. "Java The Complete Reference", Herbert Schildt, The McGraw Hill, 13th Edition, 2024
2. "Programming with Java A primer", E Balagurusamy, Tata McGraw Hill, 4th Edition, 2018
3. "J2EE Complete Reference", Jim Keogh, Tata McGraw Hill, Enterprise Edition, 2018

Semester: II

Course Name: **DATA STRUCTURE AND ALGORITHMS**

Course Code	MMC203	CIE Marks	50
Teaching Hours/Week (L:P:T)	4:0:0	SEE Marks	50
Total Hours of Pedagogy	50 hours	Total Marks	100
Credits	04	Exam Hours	03

Course Objectives:

1. Define basic data structures and explain the concepts of time and space complexity in algorithms
2. Describe and illustrate the working of fundamental data structures such as stacks, queues, and linked lists
3. Use tree and graph data structures to solve real-world problems effectively
4. Implement and evaluate algorithms for sorting and searching to address problem-solving scenarios
5. Develop real-world applications using advanced data structures and algorithm design techniques

Module – 1

FUNDAMENTALS

Basic Concepts: Definition and importance of data structures, Abstract Data Types (ADTs), Algorithm analysis: Time and space complexity, Big O notation**.**

Arrays: Definition and operations: Insertion, deletion, traversal, Multidimensional arrays, Applications of arrays.

Recursion: Definition and Principles of Recursion, Recursive Algorithms: Factorial, Fibonacci Series, Tower of Hanoi, Analysis of Recursive Algorithms.

10 Hours

MODULE – 2

STACKS, QUEUES, AND LINKED LISTS

Stacks: Definition and Operations, Applications: Conversion from Infix to Postfix Expression, Expression Evaluation, Function Calls.

Queues: Definition and Operations: Enqueue, Dequeue, Front, Rear, Types: Circular Queue, Priority Queue, Double-Ended Queue (Dequeue), Applications of Queues.

Linked Lists: Singly linked list: Creation, insertion, deletion, traversal, Doubly linked list and circular linked list, Applications of linked lists.

10 Hours

Module – 3

TREES AND GRAPHS

Trees: Definition and terminology: Root, leaf, internal node, height, depth, Binary trees: Traversal (preorder, inorder, postorder), creation, insertion, deletion, Binary search trees (BST), AVL trees, B-trees.

Graphs: Definition and terminology: Vertices, edges, adjacency list, adjacency matrix, Graph

traversal algorithms: Depth-first search (DFS), breadth-first search (BFS), Shortest path algorithm – Dijkstra's algorithm, Floyd-Warshall algorithm.

10 Hours

Module – 4

SORTING AND SEARCHING ALGORITHMS

Sorting Algorithms: Basic concepts and classification, Comparison-based sorting: Bubble sort, Selection sort, Insertion sort, Quicksort, Mergesort, Heapsort, Non-comparison-based sorting: Radix sort, counting sort.

Searching Algorithms: Linear search and Binary search, Search in linked lists, trees, and graphs, Hashing: Hash functions, collision resolution techniques (chaining, open addressing).

10 Hours

Module – 5

ADVANCED DATA STRUCTURES AND APPLICATIONS

Heaps: Definition, operations, heap sort, applications, Trie: Definition, operations, applications in dictionary and spell-checking, Segment trees and Fenwick trees: Definition, operations, range queries.

Algorithm Design Techniques**:** Divide and conquer, Greedy algorithms, Dynamic programming.

Industry Applications: Real- world applications of data structures and algorithms, Best practices in data structure and algorithm implementation, Case studies of complex problem-solving using advanced data structures.

10 Hours

Course Outcomes:

The students should be able to:	
CO1	Explain the fundamental concepts of data structures and evaluate the time and space complexity of algorithms
CO2	Demonstrate the working principles of basic data structures such as stacks, queues, and linked lists
CO3	Apply tree and graph data structures to solve real-world problems
CO4	Implement sorting and searching algorithms to address computational challenges
CO5	Design and develop real-world applications using advanced data structures and algorithmic techniques

Reference Books:

1. "Fundamentals of Data Structures in C", Ellis Horowitz and Sartaj Sahni, Universities Press, 2nd Edition, 2014
2. "Data Structures Schaum's Outlines", Seymour Lipschutz, McGraw Hill, 1st Edition, 2014
3. "Data Structures using C", Reema Thareja, Oxford press, 3rd Edition, 2012

Semester: II

Course Name: **SOFTWARE ENGINEERING**

Course Code	MMC204	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Understand the importance of various Software Engineering Lifecycle models.
2. Demonstrates agility in solving software and system challenges with a comprehensive set of skills appropriate to the needs of the dynamic global computing-based society.
3. Gain knowledge of the System Analysis and Design concepts using UML.
4. Understanding of software testing approaches such as unit testing and integration testing.
5. Understanding of the role of project management including planning, scheduling, risk management, etc.

Module – 1

SOFTWARE PROCESS AND AGILE DEVELOPMENT

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models, Introduction to Agility - Agile process, Extreme programming - XP Process, Case Study.

8 Hours

Module – 2

REQUIREMENT ENGINEERING

Functional and non – functional requirements, the software requirement document, software requirement specification, software engineering processes, requirement elicitation and analysis, requirement validation, requirement management.

8 Hours

Module – 3

SYSTEM DESIGN AND PATTERNS

Object modelling using UML, Use case Model, Class diagrams, Interaction diagrams, Activity diagrams, State chart diagrams, Functional modelling, Data Flow Diagram, CASE Tools.
Software design, Design process, Design concepts, Coupling, Cohesion, Functional independence, Design patterns tools, Model view controller, Publish, subscribe, Adapter, Command Strategy, Observer, Proxy, Facade, Architectural styles, Layered, Client Server, Tiered, Pipe and filter User interface design, Case Study.

8 Hours

Module – 4

TESTING AND MAINTENANCE

Testing- Verification and validation, Unit testing, Black box testing, White box testing, Model based testing, Integration and System testing, Regression testing, Debugging, Program analysis Symbolic execution Model Checking Case Study.

8 Hours

Module – 5

PROJECT MANAGEMENT Software Project Management, Risk Management, Software Pricing, Project Scheduling. **QUALITY MANAGEMENT** Software Quality, Software standards, reviews and inspection, Software Measurement and Metrics. **8 Hours**

Course Outcomes:

The students should be able to:	
CO1	Understand software engineering principles involved in building large software programs.
CO2	Understand the concepts of requirement specification
CO3	Design of UML, class models, design patterns and architectural styles to construct software systems.
CO4	Utilize the software testing methodologies and recognize the importance of software maintenance.
CO5	Apply estimation techniques, schedule project activities and compute pricing.

Reference Books:

1. "Software Engineering", Ian Sommerville, Pearson Education Ltd, 10th Edition, 2015
2. "Software Engineering", Pankaj jalote, Wiley India Pvt Ltd , 2010
3. "Object oriented software engineering", Stephan R. Schach, Tata McGraw Hil, 2008

Semester: II

Course Name: **WEB APPLICATION DEVELOPMENT**

Course Code	MMC205	CIE Marks	50
Teaching Hours/Week (L:P:T)	3:0:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	03	Exam Hours	03

Course Objectives:

1. Understand the fundamentals of web development, including HTML5 elements and attributes
2. Explore and illustrate the use of CSS3 selectors and principles of responsive web design
3. Describe advanced JavaScript concepts and the utilization of JavaScript libraries
4. Apply AngularJS expressions, controllers, and events to develop solutions for real-world applications
5. Demonstrate server-side programming by integrating PHP with MySQL

Module – 1

INTRODUCTION TO WEB DEVELOPMENT AND HTML5

Web Development Basics: Introduction to web technologies and protocols, Client-server architecture, Overview of front-end and back-end development

HTML5 Fundamentals: HTML5 elements and attributes, Semantic HTML5 tags, Forms and input types, Multimedia elements (audio, video)

Advanced HTML5: Canvas and SVG for graphics, HTML5 APIs (Geolocation, Web Storage, Web Workers), Offline web applications using AppCache.

8 Hours

Module – 2

CSS3 AND RESPONSIVE WEB DESIGN

CSS3 Basics: Introduction to CSS3, Selectors, properties, and values, Box model, layout, and positioning, Flexbox and Grid layouts.

Responsive Web Design: Media queries, Responsive design principles, Fluid grids and flexible images, Mobile-first design approach.

CSS Frameworks: Introduction to Bootstrap, Bootstrap components and utilities, Customizing Bootstrap with Sass (Syntactically Awesome Style Sheets).

8 Hours

Module – 3

JAVASCRIPT AND DOM MANIPULATION

Document Object Model (DOM): DOM structure and manipulation, Event handling and event listeners, Creating and modifying DOM elements, Form validation using JavaScript

Advanced JavaScript: Asynchronous JavaScript (callbacks, promises, async/await), AJAX and Fetch API, Introduction to JavaScript libraries (e.g., jQuery).

8 Hours

Module – 4

FRONT-END FRAMEWORKS AND ANGULARJS

Introduction to Front-End Frameworks: Importance of front-end frameworks, Overview of popular frameworks (React, Angular, Vue)

AngularJS Basics: Introduction to AngularJS, Modules, controllers, and scope, Directives, expressions, and filters, Form validations, Services, Application, Converting functions, Comparing functions, Events.

Advanced AngularJS: Services and dependency injection, Routing and single-page applications (SPAs), Data binding and form handling, Custom directives and components.

8 Hours

Module – 5

BACK-END INTEGRATION AND DEPLOYMENT

Back-End Development: Introduction to server-side programming, Overview of server-side languages (Node.js, PHP, Python), RESTful web services and APIs, Database integration (SQL, NoSQL)

Advanced PHP: Get and Post methods in PHP, Form processing, Uploading files, OOPs, MySQL Database – Introduction, Database connection, CRUD operations.

Full-Stack Development: Integrating front-end and back-end technologies, Developing full- stack web applications, Case studies on full-stack applications

8 Hours

Course Outcomes:

The students should be able to:	
CO1	Explain the foundational concepts of web development and the use of HTML5 elements and attributes
CO2	Demonstrate the application of CSS3 selectors and responsive web design principles
CO3	Illustrate advanced JavaScript concepts and effectively use JavaScript libraries
CO4	Develop real-world applications using AngularJS expressions, controllers, and events
CO5	Implement server-side programming by integrating PHP and MySQL to build dynamic web applications

Reference Books:

1.	"Programming the World Wide Web", Robert W. Sebesta, Pearson Education, 8th edition, 2020
2.	"Learn Angular JS in 1 Day: Complete Angular JS Guide with Examples", Krishna Rungta, Amazon Digital Services LLC, 2018
3.	"HTML5 Black Book", DT Editorial Services, Dreamtech, 2nd Edition, 2016

Semester: II

Course Name: **OBJECT ORIENTED PROGRAMMING USING JAVA LABORATORY**

Course Code	MMCL206	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:4:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	02	Exam Hours	03

Course Objectives:

1. Demonstrate object-oriented principles
2. Develop java program using data abstraction and data hiding
3. Demonstrate java program using inheritance for a given problem
4. Demonstrate java program using multithreaded concept
5. Design and develop GUI application using swings

Lab Component:

1. A team of two or four students must develop a course project using JAVA programming constructs learnt in the theory which should include the following concepts
   - Loop control structures
   - Classes and objects
   - Inheritance and polymorphism
   - Packages and generics
   - Interfaces and wrapper classes
   - Multi-threading
   - Keyboard Event and Mouse Events
   - Connection to JDBC-ODBC Bridge
2. The team must submit a brief project report (20-25 pages) that must include the following
   vi. Introduction
   vii. Requirement Analysis
   viii. Software Requirement Specification
   ix. Design and Implementation
   x. Validation
3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It was recommended that students to do prior art search as part of literature survey before submitting the synopsis for the Course-project projects.
4. Rubrics may be used to evaluate the Course-Project

Course Outcomes:

The students should be able to:	
CO1	Understand the basics of object-oriented programming using JAVA.
CO2	Apply the concept of classes, Java, JDK Components and develop Simple Java Programs.
CO3	Develop Simple Java Programs using inheritance and Exception handling.
CO4	Develop Simple Java Programs using inheritance and Exception handling.
CO5	Develop GUI applications using Swing components and Event handling programs.

Semester: II

Course Name: **DATA STRUCTURE AND ALGORITHMS LABORATORY**

Course Code	MMCL207	CIE Marks	50
Teaching Hours/Week (L:P:T)	0:4:0	SEE Marks	50
Total Hours of Pedagogy	40 hours	Total Marks	100
Credits	02	Exam Hours	03

Course Objectives:

1. Understand fundamental data structures and their implementation, including array operations
2. Illustrate the functionality and applications of basic data structures such as stacks, queues, and linked lists
3. Analyze and devise solutions to problems using tree and graph data structures
4. Explore and identify appropriate sorting and searching techniques for various problem-solving scenarios
5. Demonstrate the use of advanced data structures and algorithm design techniques to create real-world applications

Lab Component:

1. A team of two or four students must develop a course project using C/C++ programming constructs learnt in the theory which should include the following concepts
   - Stack, Queues, List
   - Trees, Graphs
   - Polynomial
   - Sparse Matrix
   - Searching algorithms
   - Sorting algorithms
2. The team must submit a brief project report (20-25 pages) that must include the following
   xi. Introduction
   xii. Requirement Analysis
   xiii. Software Requirement Specification
   xiv. Design and Implementation
   xv. Validation
3. Brief synopsis not more than two pages to be submitted by the team as per the format given. It was recommended that students to do prior art search as part of literature survey before submitting the synopsis for the Course-project projects.
4. Rubrics may be used to evaluate the Course-Project

Course Outcomes:

The students should be able to:	
CO1	Explain the basic data structures and demonstrate array operations
CO2	Illustrate the working of fundamental data structures such as stacks, queues, and linked lists
CO3	Apply tree and graph data structures to solve computational and real-world problems
CO4	Evaluate and implement appropriate sorting and searching techniques for various problem scenarios
CO5	Design and develop applications using advanced data structures and algorithmic strategies

ASSESSMENT DETAILS

Professional Core Course (PCC) & Professional Elective Course (PEC)

CIE: 50 MARKS

	Components	Number	Weightage	Max. Marks
(i)	Tests (A)	3*	60%	30
(ii)	Alternate Assessment Tools (AAT) (B)	3	40%	20
	Total Marks			50

Final CIE Marks = (A) + (B)

The following are the Alternate Assessment Tools and not limited to: Quiz, Assignments, Presentations, Paper Publications, MOOCs, Industrial Visits and Report Writing, Open Book, Self E-Learning with Certifications, Mini project and other cooperative and problem-based learning.

SEE:

The semester end exam question paper will be set for 100 marks and the marks scored by the student will be proportionately reduced to 50.

- The question paper will have ten full questions carrying equal marks.
- Each full question carries 20 marks
- There will be two full questions (with a maximum of 3 sub questions) from each module.
- Each full question will have sub question covering all the topics under a module.
- The students will have to answer five full questions selecting one full question from each module.

Integrated Professional Core Course (IPCC)

Assessment Details
CIE: 50 MARKS

	Components	Number	Weightage		Max. Marks
(i)	Tests (A)	3*	60%	60% of 30	18
(ii)	Alternate Assessment Tools (AAT) (B)	1		40% of 30	12
(iii)	Lab Component (C)	3-4	40%		20
	Total Marks				50

Final CIE Marks = (A) + (B) + (C)

The following are the Alternate Assessment Tools and not limited to: Quiz, Assignments, Presentations, Paper Publications, MOOCs, Industrial Visits and Report Writing, Open Book, Self E-Learning with Certifications, Mini project and other cooperative and problem-based learning.

SEE:

The semester end exam question paper will be set for 100 marks and the marks scored by the student will be proportionately reduced to 50.

- The question paper will have ten full questions carrying equal marks.
- Each full question carries 20 marks
- There will be two full questions (with a maximum of 3 sub questions) from each module.
- Each full question will have sub question covering all the topics under a module.
- The students will have to answer five full questions selecting one full question from each module.

Professional Core Course Lab (PCCL)

Continuous Internal Evaluation (CIE):

	Components	Weightage	Max. Marks
(i)	Lab Work: Conduction of Experiments (A)	40%	20
(ii)	Lab Journal Writing & Submission (B)	10%	05
(iii)	Lab Test (C)	30%	15
(iv)	Open-Ended Experiments (D)	20%	10
	Total Marks		50

Semester End Evaluation (SEE):

1. All laboratory experiments are to be included for practical examination
2. Students can pick one experiment from the questions lot with equal choice to all the students in a batch.
3. Change of experiment is allowed only once and 15% marks allotted to the procedure part to be made zero.
4. Marks distribution: procedure (15%) + Execution (70%) + viva voce (15%)