

Mahakaushal University, Jabalpur (M.P.)



**Master of Computer Applications (MCA)
Syllabus and Scheme of Examination**

2021-22

**Duration of Course: 2 Years
Examination Mode: Semester**

MCA 1st Semester Syllabus

MMCA0101-T (Computer Organization and Architecture)

Unit	Topic	Proposed Lectures
I	Introduction to Computer Organization & Architecture <ul style="list-style-type: none">• Overview of Computer Systems: Components and Functioning• Von Neumann Architecture vs Harvard Architecture	12
	<ul style="list-style-type: none">• Instruction Set Architecture (ISA) and Machine Instructions• Data Representation: Binary, Octal, Hexadecimal, Signed Numbers, Floating Point Representation• Computer Performance Metrics	
II	Processor and ALU Design <ul style="list-style-type: none">• CPU Organization: Control Unit, ALU, Registers, and Buses• Instruction Cycle: Fetch, Decode, Execute	12
	<ul style="list-style-type: none">• Arithmetic Operations: Addition, Subtraction, Multiplication, Division• Logic Operations and Boolean Algebra• Design of ALU, Combinational and Sequential Circuits	
III	Memory Organization <ul style="list-style-type: none">• Memory Hierarchy: Cache, Main Memory, Secondary Storage• RAM, ROM, PROM, EPROM, EEPROM	12
	<ul style="list-style-type: none">• Cache Memory: Mapping Techniques, Replacement Policies, Write Policies• Virtual Memory and Memory Management Techniques• Addressing Modes	
IV	Input/Output Organization <ul style="list-style-type: none">• I/O Interface and Data Transfer Techniques• Programmed I/O, Interrupt-Driven I/O, Direct Memory Access (DMA)	12
	<ul style="list-style-type: none">• I/O Devices: Keyboard, Mouse, Printers, Display Devices• Bus Structures: Single Bus, Multiplexed Bus• Peripheral Interfacing Basics	
V	Advanced Architecture Concepts <ul style="list-style-type: none">• Pipelining: Instruction Pipelining, Data Hazards, Control Hazards, Pipeline Stall• RISC vs CISC Architecture	12
	<ul style="list-style-type: none">• Multiprocessors and Parallel Processing Concepts• Performance Enhancement Techniques• Case Studies: Intel x86, ARM Architecture Overview	

Recommended Textbooks

1. **Carl Hamacher, Zvonko Vranesic, Safwat Zaky** – *Computer Organization*, 5th Edition, McGraw Hill
2. **David A. Patterson & John L. Hennessy** – *Computer Organization and Design: The Hardware/Software Interface*, 5th Edition, Morgan Kaufmann
3. **William Stallings** – *Computer Organization and Architecture: Designing for Performance*, 10th Edition, Pearson
4. **M. Morris Mano** – *Computer System Architecture*, 5th Edition, Pearson

Reference Books

5. **John P. Hayes** – *Computer Architecture and Organization*, McGraw Hill
6. **J. Stanley Warford** – *Computer Architecture and Organization*, McGraw Hill
7. **V. Carl Hamacher** – *Computer Organization*, McGraw Hill
8. **Barry Wilkinson & Mike Tooley** – *Digital Electronics and Computer Architecture*, Elsevier

MCA 1st Semester Syllabus

MMCA0102-T(Programming Using C)

Unit	Topic	Proposed Lectures
I	Introduction to C Language <ul style="list-style-type: none">• Overview of C Language, History and Features• Structure of a C Program, Compilation and Execution Process• Data Types, Variables, Constants, and Operators• Input/Output Functions: printf(), scanf(), getchar(), putchar()• Control Statements: if, if-else, switch-case, loops (for, while, do-while)	12
	Functions and Recursion <ul style="list-style-type: none">• Defining and Calling Functions, Function Prototypes• Pass by Value and Pass by Reference• Recursion: Factorial, Fibonacci, GCD, and Other Examples• Storage Classes: auto, extern, static, register• Scope and Lifetime of Variables	12
III	Arrays, Strings, and Pointers <ul style="list-style-type: none">• One-Dimensional and Multi-Dimensional Arrays: Declaration, Initialization, and Operations• String Handling: Declaration, Initialization, Standard Library Functions (strcpy, strlen, strcmp, etc.)• Pointers: Declaration, Initialization, Pointer Arithmetic• Pointers and Arrays, Pointers and Functions• Dynamic Memory Allocation: malloc(), calloc(), realloc(), free()	12
	Structures, Unions, and File Handling <ul style="list-style-type: none">• Structures: Definition, Declaration, Initialization, Nested Structures, Arrays of Structures• Unions: Definition and Uses• Enumerations and Typedef• File Operations: Opening, Closing, Reading, Writing, Random Access Files• Command-Line Arguments	12
V	Advanced C Concepts <ul style="list-style-type: none">• Preprocessor Directives: #define, #include, Macros, Conditional Compilation• Bitwise Operators and Manipulation• Multi-File Programs and Header Files• Error Handling in C Programs• Case Studies and Applications of C in Real-World Problems	12

Recommended Textbooks

1. **Brian W. Kernighan & Dennis M. Ritchie** – *The C Programming Language*, 2nd Edition, Prentice Hall
2. **E. Balagurusamy** – *Programming in ANSI C*, 7th Edition, McGraw Hill
3. **Yashavant Kanetkar** – *Let Us C*, 17th Edition, BPB Publications
4. **Reema Thareja** – *Programming in C*, Oxford University Press

Reference Books

5. **Deitel & Deitel** – *C How to Program*, 7th Edition, Pearson
6. **K. N. King** – *C Programming: A Modern Approach*, 2nd Edition, W. W. Norton & Company
7. **Herbert Schildt** – *C: The Complete Reference*, 2nd Edition, McGraw Hill
8. **Ashok N. Kamthane** – *Programming in C*, Pearson

MCA 1st Semester Syllabus

MMCA0102-P(C Programming Lab)

□ Basic Input/Output Operations

- Learn to handle user input and output using scanf and printf.
- Implement programs that read user input, display formatted output, and perform basic operations like addition, subtraction, etc.

□ Control Structures: if-else, switch, loops

- Implement programs using decision-making structures (if, else, switch) and loop constructs (for, while, do-while).
- Create programs to solve problems like checking even/odd, finding factorials, and generating patterns.

□ Functions and Recursion

- Understand how to create functions and call them.
- Implement recursive functions for tasks like calculating factorials, Fibonacci series, and tower of Hanoi.

□ Arrays and Multi-dimensional Arrays

- Implement programs to handle one-dimensional and two-dimensional arrays.
- Write code to perform sorting, searching, and matrix operations like addition, multiplication, and transposition.

□ String Manipulation

- Learn to handle strings in C using functions like strlen(), strcpy(), strcmp(), etc.
- Write programs for string reversal, palindrome checking, and word counting.

□ Pointers and Dynamic Memory Allocation

- Explore pointer concepts and how they can be used to manipulate variables.
- Learn to use malloc(), calloc(), free(), and realloc() for dynamic memory allocation.

□ Structures and Unions

- Understand how to define and work with structures and unions.
- Implement programs to store and manipulate data such as student records, date structures, and employee information.

□ File Handling

- Learn how to read from and write to files using functions like fopen(), fclose(), fgetc(), fputc(), fprintf(), and fscanf().
- Create programs that read/write data to text files, manipulate file content, and perform file validation.

□ Linked Lists

- Understand the concept of dynamic memory management using linked lists.

- Implement programs that create, traverse, and manipulate singly and doubly linked lists (insertion, deletion, searching).

□ **Sorting and Searching Algorithms**

- Implement sorting algorithms like bubble sort, insertion sort, selection sort, and quicksort.
- Implement search algorithms such as linear search, binary search, and hash-based search methods.

MARKU

MCA 1st Semester Syllabus

MMCA0103-T(Data Structures)

Unit	Topic	Proposed Lectures
I	Introduction to Data Structures <ul style="list-style-type: none">• Definition, Importance, and Classification of Data Structures• Abstract Data Types (ADT)• Arrays: Single and Multidimensional, Operations• Linked Lists: Singly, Doubly, and Circular Linked Lists• Applications of Arrays and Linked Lists	12
	Stacks and Queues <ul style="list-style-type: none">• Stack ADT: Operations, Implementation (Array and Linked List)• Applications of Stack: Expression Evaluation (Infix, Prefix, Postfix), Parenthesis Matching• Queue ADT: Simple Queue, Circular Queue, Priority Queue, Dequeue• Queue Implementation using Arrays and Linked Lists	
III	Trees <ul style="list-style-type: none">• Tree Terminology: Root, Node, Height, Depth, Degree• Binary Trees: Representation, Traversals (Inorder, Preorder, Postorder)• Binary Search Trees (BST): Insertion, Deletion, Searching• Balanced Trees: AVL Trees, Rotations• Applications of Trees	12
	Graphs <ul style="list-style-type: none">• Graph Terminology: Vertices, Edges, Degree, Types of Graphs• Graph Representation: Adjacency Matrix and List• Graph Traversal Algorithms: BFS, DFS• Shortest Path Algorithms: Dijkstra, Floyd-Warshall• Minimum Spanning Tree: Kruskal's and Prim's Algorithms	
V	Advanced Data Structures and Applications <ul style="list-style-type: none">• Heaps: Min Heap, Max Heap, Heap Operations• Hashing: Hash Functions, Collision Resolution Techniques (Chaining, Open Addressing)• Searching and Sorting Techniques: Linear Search, Binary Search, Quick Sort, Merge Sort, Heap Sort• Applications of Advanced Data Structures in Real-World Problems	12

Recommended Textbooks

1. **Horowitz, Sahni & Anderson-Freed** – *Fundamentals of Data Structures in C*, 2nd Edition, Universities Press
2. **S. Lipschutz** – *Data Structures*, Schaum's Outline Series, McGraw Hill
3. **Robert Lafore** – *Data Structures and Algorithms in Java*, 2nd Edition, Sams Publishing
4. **Mark Allen Weiss** – *Data Structures and Algorithm Analysis in C++*, 4th Edition, Pearson

Reference Books

5. **Narasimha Karumanchi** – *Data Structures and Algorithms Made Easy*, CareerMonk Publications
6. **Adam Drozdek** – *Data Structures and Algorithms in C++*, 4th Edition, Cengage Learning
7. **Tanenbaum, Langsam & Augenstein** – *Data Structures Using C*, Pearson
8. **E. Horowitz & S. Sahni** – *Fundamentals of Data Structures*, Computer Science Press

MMCA0103-P(Data Structure Lab)

□ Array Operations

- Implement basic array operations such as insertion, deletion, searching, and sorting.
- Implement algorithms like bubble sort, selection sort, and linear/binary search for arrays.

□ Linked List Operations

- Create programs to perform operations on singly linked lists like insertion (at beginning, end, specific position), deletion, and traversal.
- Implement doubly linked lists and circular linked lists.

□ Stack Implementation

- Implement a stack using arrays and linked lists.
- Perform stack operations like push, pop, peek, and isEmpty.
- Solve problems using stacks such as expression evaluation (infix to postfix conversion, postfix evaluation).

□ Queue Implementation

- Implement a queue using arrays and linked lists.
- Perform operations like enqueue, dequeue, front, rear, and isEmpty.
- Implement circular queues and priority queues.
- Use queues for applications such as scheduling and simulation problems.

□ Binary Search Tree (BST)

- Implement operations on binary search trees, including insertion, deletion, search, and in-order traversal.
- Write algorithms for tree traversal (pre-order, in-order, post-order).
- Implement functions for finding the minimum and maximum values in a BST, and checking if the tree is balanced.

□ Heap (Min-Heap and Max-Heap)

- Implement a binary heap for both min-heap and max-heap.
- Implement heap operations such as insert, delete, and heapify.
- Use heaps for sorting (heap sort) and priority queue implementation.

□ Graph Representation and Traversal

- Implement a graph using adjacency matrix and adjacency list.
- Perform graph traversal using Depth First Search (DFS) and Breadth First Search (BFS).
- Solve problems like finding connected components or shortest path using BFS.

□ Hashing and Hash Tables

- Implement a hash table with collision resolution techniques such as linear probing, quadratic probing, and separate chaining.
- Create functions for insertion, deletion, and searching in a hash table.
- Implement a simple application, such as a dictionary or contact management system, using a hash table.

□ Trie (Prefix Tree)

- Implement a trie for efficient string storage and retrieval.
- Implement operations like insertion, searching, and deletion in a trie.
- Use tries to solve problems like autocomplete or prefix matching.

□ Graph Algorithms

- Implement Dijkstra's algorithm for shortest path finding.
- Implement Bellman-Ford algorithm for detecting negative weight cycles.
- Implement the Floyd-Warshall algorithm for finding shortest paths between all pairs of nodes.
- Solve problems like finding the minimum spanning tree using Prim's or Kruskal's algorithm.

MCA 1st Semester Syllabus

MMCA0104-T(Discrete Mathematics)

Unit	Topic	Proposed Lectures
I	Fundamentals of Logic and Proof Techniques	
	• Statements, Logical Connectives, Truth Tables	
	• Propositional Equivalences, Implications, and Logical Arguments	12
	• Predicate Logic: Predicates, Quantifiers, and Nested Quantifiers	
	• Methods of Proof: Direct, Indirect, Contradiction, Contrapositive, Mathematical Induction	
II	• Applications of Logic in Computer Science	
	Set Theory and Functions	
	• Sets, Subsets, Power Sets, Operations on Sets	
	• Venn Diagrams, Cartesian Products	12
	• Relations: Properties, Equivalence Relations, Partial Ordering	
III	• Functions: One-to-One, Onto, Bijective, Inverse Functions, Composition	
	• Applications of Set Theory in Computing	
	Combinatorics and Counting	
	• Basic Counting Principles: Addition and Multiplication Rules	
	• Permutations and Combinations	12
IV	• Binomial Theorem and Pascal's Triangle	
	• Pigeonhole Principle	
	• Applications of Counting in Problem Solving	
	Graph Theory	
	• Basic Concepts: Graphs, Vertices, Edges, Degree	
V	• Types of Graphs: Simple, Directed, Weighted, Bipartite, Complete, Trees	
	• Graph Representation: Adjacency Matrix/List	12
	• Graph Traversals: BFS, DFS	
	• Applications of Graphs in Networks, Scheduling, and Optimization	
	Algebraic Structures and Recurrence Relations	
V	• Algebraic Structures: Groups, Semi-Groups, Monoids, Rings, Fields (Basic Concepts)	
	• Recurrence Relations: Linear, Non-Linear, Homogeneous, Non-Homogeneous	12
	• Solving Recurrence Relations Using Characteristic Equation	
	• Generating Functions	
	• Applications in Algorithm Analysis and Computer Science Problems	

Recommended Textbooks

1. **Kenneth H. Rosen** – *Discrete Mathematics and Its Applications*, 8th Edition, McGraw Hill
2. **R. P. Grimaldi** – *Discrete and Combinatorial Mathematics*, 5th Edition, Pearson
3. **C.L. Liu & D.P. Mohapatra** – *Elements of Discrete Mathematics*, McGraw Hill
4. **Susanna S. Epp** – *Discrete Mathematics with Applications*, Cengage

Reference Books

5. **S. Lipschutz & M. Lipson** – *Discrete Mathematics*, McGraw Hill
6. **J. L. Gross & J. Yellen** – *Graph Theory and Its Applications*, CRC Press
7. **Joe L. Mott, Abraham Kandel, Theodore P. Baker** – *Discrete Mathematics for Computer Scientists*, Pearson
8. **Bernard Kolman & Robert C. Busby** – *Discrete Mathematical Structures*, Pearson

MCA 1st Semester Syllabus

MMCA0105-T(Operating System)

Unit	Topic	Proposed Lectures
I	Introduction to Operating Systems <ul style="list-style-type: none">• Definition, Objectives, and Functions of an Operating System• Types of Operating Systems: Batch, Multiprogramming, Time-Sharing, Real-Time, Distributed• OS Services and User Interface• System Calls, OS Architecture: Monolithic, Layered, Microkernel• Operating System Structure and Components	12
	Process Management <ul style="list-style-type: none">• Process Concept, Process Control Block (PCB)• Process States and State Transition Diagrams• Threads: Types, Advantages, and Models• Process Scheduling: Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, Priority, Round Robin)• Multilevel Queue and Multilevel Feedback Queue Scheduling	12
III	Synchronization and Deadlocks <ul style="list-style-type: none">• Critical Section Problem• Semaphores, Mutexes, Monitors• Classical Synchronization Problems: Producer-Consumer, Readers-Writers, Dining Philosophers• Deadlock: Characterization, Necessary Conditions, Deadlock Prevention, Avoidance, Detection, and Recovery	12
IV	Memory Management <ul style="list-style-type: none">• Logical vs Physical Address Space, Contiguous Memory Allocation• Paging, Segmentation• Virtual Memory: Concepts, Demand Paging, Page Replacement Algorithms (FIFO, LRU, Optimal)• Thrashing and Working Set Model• Memory Management in Modern Operating Systems	12
V	File Systems and I/O Management <ul style="list-style-type: none">• File Concept, File Types, File Operations, File System Structure• Directory Structure, File Allocation Methods (Contiguous, Linked, Indexed)• Disk Scheduling Algorithms: FCFS, SSTF, SCAN, C-SCAN• RAID Levels, Disk Management, I/O Systems• Case Studies: Unix/Linux and Windows OS Overview	12

Recommended Textbooks

1. **Abraham Silberschatz, Peter B. Galvin, Greg Gagne** – *Operating System Concepts*, 10th Edition, Wiley
2. **William Stallings** – *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson
3. **Harsh B. Patel** – *Operating Systems Concepts*, Oxford University Press
4. **D.M. Dhamdhere** – *Operating Systems: A Concept-Based Approach*, 2nd Edition, McGraw Hill

Reference Books

5. **Tanenbaum & Bos** – *Modern Operating Systems*, 4th Edition, Pearson
6. **Gary Nutt** – *Operating Systems: A Modern Perspective*, 3rd Edition, Pearson
7. **Charles Crowley** – *Operating Systems: A Design-Oriented Approach*, McGraw Hill

8. **Rajib Mall** – *Operating System: Concepts and Design*, PHI Learning

 **Online Resources**

- **NPTEL – Operating Systems:** <https://nptel.ac.in/>
- **MIT OpenCourseWare – Operating Systems:** <https://ocw.mit.edu/courses/6-828-operating-system-engineering-fall-2012/>
- **GeeksforGeeks – Operating System Tutorials:** <https://www.geeksforgeeks.org/operating-systems/>
- **Tutorialspoint – OS Concepts:** https://www.tutorialspoint.com/operating_system/index.htm

MARKU

MCA 2nd Semester Syllabus

MMCA0201-T(Java Programming)

Unit	Topic	Proposed Lectures			
I	Introduction to Java <ul style="list-style-type: none">• Features of Java and JVM Architecture• Java Development Environment and JDK Setup• Data Types, Variables, Constants, and Operators• Control Statements: if-else, switch, loops (for, while, do-while)• Introduction to Classes and Objects, Methods	12			
	Object-Oriented Concepts in Java <ul style="list-style-type: none">• Classes, Objects, and Methods• Constructors and Destructor Concepts (Finalize Method)		12		
	II <ul style="list-style-type: none">• Inheritance: Single, Multilevel, Hierarchical, and Hybrid• Method Overloading and Overriding• Polymorphism and Encapsulation• Packages and Access Specifiers			12	
	Advanced Java Features <ul style="list-style-type: none">• Interfaces and Abstract Classes• Exception Handling: try, catch, finally, throw, throws, custom exceptions• String Handling and StringBuffer/StringBuilder Classes• Wrapper Classes and Autoboxing/Unboxing• Enumerations and Annotations				12
	III <ul style="list-style-type: none">• Collections and Multithreading• Introduction to Java Collections Framework (List, Set, Map, Queue)• Iterators and Enhanced for Loop• Generics in Java• Multithreading Concepts: Thread Class, Runnable Interface• Thread Synchronization, Deadlocks, and Thread Communication				
IV <ul style="list-style-type: none">• GUI Programming and I/O in Java• Applets and Swing Components: JFrame, JButton, JLabel, JTextField, JTextArea, JPanel• Event Handling: ActionListener, MouseListener, KeyListener	12				
V <ul style="list-style-type: none">• Layout Managers: BorderLayout, GridLayout, FlowLayout• Java I/O Streams: File, FileReader/FileWriter, BufferedReader/BufferedWriter, Object Serialization• Introduction to JDBC and Database Connectivity		12			

Recommended Textbooks

1. **Herbert Schildt** – *Java: The Complete Reference*, 11th Edition, McGraw Hill
2. **Cay S. Horstmann** – *Core Java Volume I – Fundamentals*, 11th Edition, Pearson
3. **E. Balagurusamy** – *Programming with Java*, McGraw Hill
4. **Paul Deitel & Harvey Deitel** – *Java How to Program*, 11th Edition, Pearson

Reference Books

5. **Bruce Eckel** – *Thinking in Java*, 4th Edition, Prentice Hall
6. **K. Arnold & J. Gosling** – *The Java Programming Language*, 4th Edition, Addison-Wesley
7. **Herbert Schildt** – *Java 8 Programming Black Book*, Dreamtech Press
8. **Y. Daniel Liang** – *Introduction to Java Programming*, 11th Edition, Pearson

MMCA0201-P(Java Programming Lab)

Practical 1: File Handling and Serialization

- **Objective:** Perform file operations using Java I/O streams and demonstrate serialization.
- **Task:**
 - Create a program that reads from and writes to files using FileReader, FileWriter, and BufferedReader.
 - Implement object serialization: Create a class, serialize an object of that class to a file, and then deserialize it.

Practical 2: Implementing Collections Framework

- **Objective:** Use Java's Collection Framework (List, Set, Map).
- **Task:**
 - Implement a program to use different collections like ArrayList, HashSet, and HashMap.
 - Perform operations like adding, removing, iterating, and searching elements within these collections.

Practical 3: Multi-threading with Synchronization

- **Objective:** Create multithreaded applications and understand synchronization.
- **Task:**
 - Create two threads that perform different tasks (e.g., reading and writing data).
 - Use synchronized blocks/methods to prevent race conditions and ensure thread safety.

Practical 4: Thread Communication Using wait() and notify()

- **Objective:** Demonstrate inter-thread communication using wait(), notify(), and notifyAll().
- **Task:**
 - Create two threads: one producer and one consumer.
 - Use wait() and notify() to handle the production and consumption of data.

Practical 5: Java Socket Programming

- **Objective:** Create client-server applications using Java sockets.
- **Task:**
 - Develop a simple client-server application where the client sends a message to the server, and the server responds with a confirmation message.
 - Use Socket and ServerSocket classes to establish a connection.

Practical 6: Java Database Connectivity (JDBC)

- **Objective:** Establish a connection between Java and a relational database.
- **Task:**
 - Write a program to connect to a MySQL/Oracle database using JDBC.
 - Implement operations like inserting, updating, deleting, and retrieving data from a database table.

Practical 7: Implementing a Simple Chat Application (Using Sockets)

- **Objective:** Develop a basic chat application using Java Sockets.
- **Task:**
 - Create a server and multiple clients using Socket and ServerSocket.
 - Allow clients to send messages to the server and receive messages from other clients.

Practical 8: Java Remote Method Invocation (RMI)

- **Objective:** Implement RMI to enable communication between Java programs on different machines.
- **Task:**
 - Create an RMI application that involves a server that provides services (e.g., simple calculation) and clients that access those services remotely.
 - Use Remote, UnicastRemoteObject, and the RMI Registry.

Practical 9: Java Servlet and JSP

- **Objective:** Create dynamic web applications using Java Servlets and JSP.
- **Task:**
 - Write a Java Servlet to process HTTP requests and generate dynamic responses.
 - Create a simple web application using JSP to display dynamic data from a backend database.

Practical 10: Java NIO (New Input/Output)

- **Objective:** Use Java NIO for non-blocking I/O operations.
- **Task:**
 - Create a program using Java NIO's FileChannel, ByteBuffer, and Path classes.
 - Implement file reading/writing using NIO to compare performance with traditional Java I/O.

MCA 2nd Semester Syllabus

MMCA0202-T(Database Management System)

Unit	Topic	Proposed Lectures				
I	Introduction to DBMS <ul style="list-style-type: none">• Overview of Data, Database, and Database Management Systems• Advantages and Disadvantages of DBMS over File Systems• DBMS Architecture: Three-Tier Architecture, DBMS Components• Database Models: Hierarchical, Network, Relational, Object-Oriented• Database Users: DBA, End Users, and Application Programmers	12				
	II		Relational Database Concepts <ul style="list-style-type: none">• Relations, Attributes, Tuples, and Keys• Relational Algebra: Select, Project, Join, Union, Intersection, Difference• Relational Calculus: Tuple and Domain Calculus• Integrity Constraints: Primary Key, Foreign Key, Domain Constraints• SQL Basics: Data Definition Language (DDL) and Data Manipulation Language (DML)	12		
			III		Advanced SQL & Query Processing <ul style="list-style-type: none">• Advanced SQL Queries: Subqueries, Joins, Views, Indexes• Aggregate Functions, Grouping, and Sorting• Transactions: Properties of Transactions (ACID)• Concurrency Control: Locks, Deadlocks, and Isolation Levels• Query Optimization Techniques	12
					IV	
V		Advanced DBMS Concepts <ul style="list-style-type: none">• Indexing and Hashing: B-Trees, B+ Trees, ISAM• Distributed Databases: Concepts and Architecture• NoSQL Databases: Types, Features, and Applications• Database Security, Backup, and Recovery• Emerging Trends: Cloud Databases, Big Data Integration, Data Warehousing				

Recommended Textbooks

1. **Raghu Ramakrishnan & Johannes Gehrke** – *Database Management Systems*, 3rd Edition, McGraw Hill
2. **Elmasri & Navathe** – *Fundamentals of Database Systems*, 7th Edition, Pearson
3. **C.J. Date** – *An Introduction to Database Systems*, 8th Edition, Pearson
4. **Abraham Silberschatz, Henry Korth, S. Sudarshan** – *Database System Concepts*, 7th Edition, McGraw Hill

Reference Books

5. **P. S. Deshpande** – *Database Management Systems*, Wiley
6. **Ramez Elmasri & Shamkant B. Navathe** – *Fundamentals of Database Systems*, Pearson
7. **Harrington** – *Relational Database Design and Implementation*, Morgan Kaufmann
8. **Jeffrey D. Ullman & Jennifer Widom** – *A First Course in Database Systems*, Pearson

MMCA0202-P(Database Management System) lab

Practical 1: SQL Queries for Data Definition

- **Objective:** Learn and practice SQL Data Definition Language (DDL) commands.
- **Task:**
 - Create a database and define tables using CREATE command.
 - Implement constraints like PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, and CHECK.
 - Alter the table structure using ALTER command.
 - Drop a table using the DROP command.

Practical 2: SQL Queries for Data Manipulation

- **Objective:** Learn and practice SQL Data Manipulation Language (DML) commands.
- **Task:**
 - Insert data into tables using INSERT INTO.
 - Update existing records using UPDATE.
 - Delete records using DELETE.
 - Retrieve data using SELECT with different conditions.

Practical 3: SQL Queries for Data Retrieval (Advanced)

- **Objective:** Practice advanced SQL queries with complex conditions.
- **Task:**
 - Use JOIN (INNER, LEFT, RIGHT, and FULL) to combine multiple tables.
 - Implement GROUP BY, HAVING, and ORDER BY clauses for aggregation and sorting.
 - Use subqueries in SELECT, INSERT, UPDATE, and DELETE operations.
 - Use UNION, INTERSECT, and EXCEPT to combine results.

Practical 4: Normalization and De-Normalization

- **Objective:** Apply normalization techniques to reduce redundancy in a relational database.
- **Task:**
 - Given a set of unnormalized relations, normalize the schema up to the 3rd normal form (3NF).
 - Identify and remove partial and transitive dependencies.
 - Demonstrate de-normalization techniques to improve performance in certain scenarios.

Practical 5: Database Design Using ER and Relational Models

- **Objective:** Design a database schema from an entity-relationship (ER) diagram.
- **Task:**
 - Create an ER diagram for a real-world system (e.g., library, hospital, or e-commerce system).
 - Convert the ER diagram into a relational schema (tables).
 - Implement the schema in the DBMS and populate it with sample data.

Practical 6: Transaction Management

- **Objective:** Understand and implement transaction management in DBMS.
- **Task:**
 - Simulate a transaction and demonstrate the properties of ACID (Atomicity, Consistency, Isolation, Durability).
 - Perform a series of transactions, including COMMIT, ROLLBACK, and SAVEPOINT.

- Use SET TRANSACTION to control transaction isolation levels (e.g., READ COMMITTED, SERIALIZABLE).

Practical 7: Indexing in Databases

- **Objective:** Implement indexing to improve query performance.
- **Task:**
 - Create primary and secondary indexes on tables.
 - Experiment with clustered and non-clustered indexes.
 - Perform query operations (e.g., SELECT, UPDATE, DELETE) on indexed and non-indexed tables and compare performance.

Practical 8: SQL Views and Triggers

- **Objective:** Use views and triggers to implement business rules and abstract data.
- **Task:**
 - Create views to simplify complex queries or limit access to sensitive data.
 - Create triggers to automate actions (e.g., updating or deleting records) based on specific events (INSERT, UPDATE, DELETE).
 - Demonstrate the use of AFTER and BEFORE triggers.

Practical 9: Database Backup and Recovery

- **Objective:** Implement backup and recovery strategies to safeguard the database.
- **Task:**
 - Perform full and incremental backups of the database.
 - Simulate a system failure and demonstrate how to recover the database using backup files.
 - Restore a database from a backup and test data integrity after recovery.

Practical 10: SQL Injection and Security

- **Objective:** Learn about SQL injection vulnerabilities and implement security measures.
- **Task:**
 - Simulate SQL injection attacks on a vulnerable application and understand its impact.
 - Implement security best practices to prevent SQL injection, such as using prepared statements, parameterized queries, and input validation.

MCA 2nd Semester Syllabus

MMCA0203-T(Scientific Computing)

Unit	Topic	Proposed Lectures	
I	Introduction to Scientific Computing <ul style="list-style-type: none">• Definition, Scope, and Applications in Engineering, Physics, and Computational Sciences• Computational Thinking and Problem Solving Strategies• Floating-Point Arithmetic and Numerical Accuracy• Errors in Scientific Computing: Round-Off, Truncation, and Propagation• Introduction to Scientific Computing Software (MATLAB, Python, R)	12	
	Numerical Methods for Linear Systems <ul style="list-style-type: none">• Direct Methods: Gaussian Elimination, LU Decomposition		12
	Numerical Methods for Nonlinear Equations & Interpolation <ul style="list-style-type: none">• Iterative Methods: Jacobi, Gauss-Seidel, Successive Over-Relaxation (SOR)• Eigenvalues and Eigenvectors Computation• Applications of Linear Systems in Modeling Physical Problems		
	Numerical Differentiation, Integration, and ODEs <ul style="list-style-type: none">• Root-Finding Techniques: Bisection, Newton-Raphson, Secant Method		12
II	Interpolation: Lagrange, Newton, and Spline Interpolation <ul style="list-style-type: none">• Approximation and Curve Fitting• Error Analysis and Convergence Criteria	12	
	Numerical Differentiation, Integration, and ODEs <ul style="list-style-type: none">• Numerical Differentiation Formulas		12
III	Numerical Integration: Trapezoidal, Simpson's 1/3 & 3/8 Rules, Gaussian Quadrature <ul style="list-style-type: none">• Solution of Ordinary Differential Equations (Euler, Modified Euler, Runge-Kutta Methods)• Applications in Physics and Engineering Problems	12	
	Advanced Topics in Scientific Computing <ul style="list-style-type: none">• Partial Differential Equations: Finite Difference Method• Optimization Techniques in Scientific Computing		12
IV	Monte Carlo Simulations <ul style="list-style-type: none">• Parallel and High-Performance Computing Concepts• Case Studies and Applications in Computational Biology, Fluid Dynamics, and Climate Modeling	12	
	Case Studies and Applications in Computational Biology, Fluid Dynamics, and Climate Modeling		

Recommended Textbooks

1. **M. K. Jain, S. R. K. Iyengar, R. K. Jain** – *Numerical Methods for Scientific and Engineering Computation*, New Age International
2. **Gerald & Wheatley** – *Applied Numerical Analysis*, Pearson
3. **C. F. Gerald & P. O. Wheatley** – *Applied Numerical Analysis*, Pearson
4. **E. Balagurusamy** – *Numerical Methods*, McGraw Hill

Reference Books

5. **K. E. Atkinson** – *An Introduction to Numerical Analysis*, Wiley
6. **S. S. Sastry** – *Introductory Methods of Numerical Analysis*, PHI Learning
7. **Richard L. Burden & J. Douglas Faires** – *Numerical Analysis*, Cengage
8. **M. C. Joshi** – *Computer Oriented Numerical Methods*, PHI Learning

MCA 2nd Semester Syllabus

MMCA0204-T(Theory of Computation)

Unit	Topic	Proposed Lectures
I	Introduction to Theory of Computation <ul style="list-style-type: none">• Fundamentals of Computation and Automata Theory• Alphabets, Strings, and Languages• Chomsky Hierarchy: Type 0, Type 1, Type 2, Type 3• Deterministic vs Non-Deterministic Computation• Applications of Automata Theory	12
	Finite Automata <ul style="list-style-type: none">• Deterministic Finite Automata (DFA): Definition, Transition Diagram, Transition Table• Non-Deterministic Finite Automata (NFA) and Equivalence with DFA• Finite Automata with Epsilon Transitions (ϵ-NFA)• Regular Languages and Regular Expressions• Closure Properties of Regular Languages and Minimization of DFA	
III	Context-Free Grammars & Pushdown Automata <ul style="list-style-type: none">• Context-Free Grammar (CFG): Definition and Derivation• Parse Trees, Leftmost and Rightmost Derivation• Ambiguity in CFGs and Simplification Techniques• Pushdown Automata (PDA): Deterministic and Non-Deterministic• Equivalence of PDA and CFG	12
	Turing Machines and Computability <ul style="list-style-type: none">• Turing Machine (TM): Definition, Components, and Example• Variants of Turing Machine: Multi-Tape, Non-Deterministic TM• Church-Turing Thesis• Decidability and Recursive Languages• Halting Problem and Undecidable Problems	
V	Advanced Topics in Computation <ul style="list-style-type: none">• Linear Bounded Automata (LBA) and Context-Sensitive Languages• Closure Properties of Context-Free and Context-Sensitive Languages• Introduction to Complexity Theory: P, NP, NP-Complete, NP-Hard• Reducibility and Examples of NP-Complete Problems• Applications of Theory of Computation in Compilers and Formal Verification	12

Recommended Textbooks

1. **John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** – *Introduction to Automata Theory, Languages, and Computation*, 3rd Edition, Pearson
2. **Peter Linz** – *An Introduction to Formal Languages and Automata*, 6th Edition, Jones & Bartlett
3. **Michael Sipser** – *Introduction to the Theory of Computation*, 3rd Edition, Cengage Learning
4. **Daniel I.A. Cohen** – *Introduction to Computer Theory*, John Wiley & Sons

Reference Books

5. **K.L.P. Mishra & N. Chandrasekaran** – *Theory of Computer Science: Automata, Languages, and Computation*, PHI Learning
6. **Eitan Gurari** – *Introduction to Computation and Automata Theory*, CRC Press
7. **D. Cohen** – *Elements of Computer Theory*, McGraw Hill
8. **H. R. Lewis & C. H. Papadimitriou** – *Elements of the Theory of Computation*, Prentice Hall

MCA 2nd Semester Syllabus

MMCA0205-T (Data Communication & Computer Networks)

Unit	Topic	Proposed Lectures
I	Introduction to Data Communication	
	• Components of Data Communication: Sender, Receiver, Medium, and Protocols	
	• Types of Data: Analog and Digital	
	• Data Flow: Simplex, Half-Duplex, Full-Duplex	12
	• Transmission Modes and Characteristics	
II	• Network Types: LAN, MAN, WAN, PAN	
	• OSI Model vs TCP/IP Model	
	Transmission Media and Switching	
	• Guided Media: Twisted Pair, Coaxial Cable, Fiber Optics	
	• Unguided Media: Radio Waves, Microwave, Satellite Communication	12
III	• Transmission Impairments: Attenuation, Noise, Distortion, Interference	
	• Switching Techniques: Circuit Switching, Packet Switching, Message Switching	
	• Multiplexing Techniques: FDM, TDM, WDM	
	Data Link Layer & Error Control	
	• Framing Techniques: Character, Bit, and Byte Oriented	
IV	• Error Detection: Parity, CRC, Checksum	12
	• Error Correction: Hamming Code, Forward Error Correction	
	• Flow Control: Stop-and-Wait, Sliding Window Protocol	
	• MAC Techniques: CSMA/CD, CSMA/CA, Token Ring, Token Bus	
	Network Layer & Transport Layer	
V	• Routing Algorithms: Static vs Dynamic, Distance Vector, Link State	
	• Congestion Control and Quality of Service (QoS)	12
	• IP Addressing: IPv4, IPv6, Subnetting	
	• Transport Layer Protocols: TCP, UDP	
	• Socket Programming Basics	
	Advanced Networking Concepts	
	• Wireless Networks: Wi-Fi, Bluetooth, Cellular Networks	
	• Network Security Basics: Cryptography, Firewalls, VPN	12
	• Network Management: SNMP, Network Monitoring Tools	
	• Emerging Trends: SDN, IoT Networks, Cloud Networking	
	• Case Studies of LAN, WAN, and Internet Applications	

Recommended Textbooks

1. **Andrew S. Tanenbaum & David J. Wetherall** – *Computer Networks*, 5th Edition, Pearson
2. **Behrouz A. Forouzan** – *Data Communications and Networking*, 5th Edition, McGraw Hill
3. **James F. Kurose & Keith W. Ross** – *Computer Networking: A Top-Down Approach*, 8th Edition, Pearson
4. **Douglas E. Comer** – *Computer Networks and Internets*, 6th Edition, Pearson

Reference Books

5. **William Stallings** – *Data and Computer Communications*, 10th Edition, Pearson
6. **Fred Halsall** – *Computer Networking and the Internet*, 5th Edition, Pearson
7. **Larry L. Peterson & Bruce S. Davie** – *Computer Networks: A Systems Approach*, 5th Edition, Morgan Kaufmann
8. **Gary M. Shelly, Thomas J. Cashman** – *Networking Essentials*, Cengage Learning

MCA 3rd Semester Syllabus

MMCA0301-T (COMPUTER ORIENTED OPTIMIZATION TECHNIQUES)

Unit	Topic	Proposed Lectures
I	Introduction to Optimization Techniques <ul style="list-style-type: none">• Definition, Scope, and Applications of Optimization in Engineering and Management• Classification of Optimization Problems: Single-Objective, Multi-Objective, Continuous, and Discrete	12
	<ul style="list-style-type: none">• General Approach to Optimization• Unconstrained vs Constrained Optimization• Role of Computers in Solving Optimization Problems	
II	Linear Programming Techniques <ul style="list-style-type: none">• Formulation of Linear Programming Problems (LPP)• Graphical Method for Two-Variable Problems	12
	<ul style="list-style-type: none">• Simplex Method: Basic Concepts, Algorithm, and Solution• Big-M and Two-Phase Methods• Duality in Linear Programming and Sensitivity Analysis	
III	Non-Linear Programming Techniques <ul style="list-style-type: none">• Unconstrained Optimization: Necessary and Sufficient Conditions for Optimality• One-Dimensional Minimization Methods: Fibonacci, Golden Section, and Quadratic Interpolation	12
	<ul style="list-style-type: none">• Multivariable Methods: Steepest Descent, Newton-Raphson, and Conjugate Gradient• Constrained Optimization: Lagrange Multipliers and Kuhn-Tucker Conditions	
IV	Dynamic Programming and Integer Optimization <ul style="list-style-type: none">• Introduction to Dynamic Programming• Bellman's Principle of Optimality	12
	<ul style="list-style-type: none">• Applications in Resource Allocation and Inventory Management• Integer and Mixed-Integer Programming• Branch and Bound Method for Integer Optimization	
V	Modern Optimization Techniques and Applications <ul style="list-style-type: none">• Transportation and Assignment Problems• Network Optimization: Shortest Path, Minimum Spanning Tree, and Max Flow	12
	<ul style="list-style-type: none">• Introduction to Heuristic Methods: Genetic Algorithms, Simulated Annealing, Particle Swarm Optimization• Multi-Criteria Decision Making (MCDM)• Case Studies and Industrial Applications of Optimization Techniques	

Recommended Textbooks

1. **K. V. Mital & C. Mohan** – *Optimization Methods in Operations Research and Systems Analysis*, New Age International
2. **Hamdy A. Taha** – *Operations Research: An Introduction*, Pearson Education
3. **S. S. Rao** – *Engineering Optimization: Theory and Practice*, Wiley
4. **F. S. Hillier & G. J. Lieberman** – *Introduction to Operations Research*, McGraw Hill

 **Reference Books**

5. **R. Panneerselvam** – *Operations Research*, Prentice Hall India
6. **J. K. Sharma** – *Operations Research: Theory and Applications*, Macmillan
7. **G. Hadley** – *Linear Programming*, Addison Wesley
8. **M. C. Joshi & P. K. Gupta** – *Computer Oriented Optimization Techniques*, PHI Learning

MARKU

MCA 3rd Semester Syllabus

MMCA0302-T(Microprocessor & Assembly Language Programming)

Unit	Topic	Proposed Lectures
I	Introduction to Microprocessors <ul style="list-style-type: none">• Overview of Microprocessors and Microcontrollers• History and Evolution of Microprocessors	12
	<ul style="list-style-type: none">• Microprocessor vs Microcontroller• Architecture of 8085 and 8086 Microprocessor• Pin Configuration and Functional Blocks• Instruction Set Architecture and Addressing Modes	
II	Assembly Language Programming <ul style="list-style-type: none">• Introduction to Assembly Language• Data Transfer, Arithmetic, and Logical Instructions	12
	<ul style="list-style-type: none">• Branching and Looping Instructions• Stack, Subroutine, and Procedure Calls• Assembler Directives and Macros• Writing and Debugging Simple Assembly Programs	
III	Timing, Control, and Interfacing <ul style="list-style-type: none">• Timing and Control Signals of Microprocessor• Memory Interfacing: ROM, RAM	12
	<ul style="list-style-type: none">• Input/Output Interfacing: Ports, Latches, Buffers• Interrupts: Types, Servicing, and Priority• Peripheral Interface Devices: 8255, 8259, 8251, 8253• Serial and Parallel Data Communication	
IV	Advanced Assembly and Programming Techniques <ul style="list-style-type: none">• Multiplication and Division Instructions• Bit Manipulation and Rotate Instructions	12
	<ul style="list-style-type: none">• String Instructions and Block Transfer• Modular Programming in Assembly• Interfacing Keyboards, Displays, ADC/DAC• Practical Examples: Traffic Light Controller, Digital Clock	
V	Applications and Modern Trends <ul style="list-style-type: none">• Microprocessor-Based Systems• Embedded System Concepts	12
	<ul style="list-style-type: none">• Microprocessor Applications in Robotics, Industrial Automation, and IoT• Introduction to 8051/ARM Microcontrollers• Comparison with Modern Microprocessors (ARM Cortex Series)• Case Studies of Real-Time Embedded Applications	

Recommended Textbooks

1. **Ramesh S. Gaonkar** – *Microprocessor Architecture, Programming and Applications with the 8085*, Penram International
2. **K. Udaya Kumar & B.S. Umashankar** – *Microprocessors and Interfacing*, TMH
3. **Douglas V. Hall** – *Microprocessors and Interfacing: Programming and Hardware*, McGraw Hill
4. **Barry B. Brey** – *The Intel Microprocessors*, Pearson

 **Reference Books**

5. **A.K. Ray & K.M. Bhurchandi** – *Advanced Microprocessors and Peripherals*, TMH
6. **John Uffenbeck** – *8085/8086 Microprocessor Programming & Interfacing*, PHI Learning
7. **Muhammad Ali Mazidi, Janice Mazidi & Rolin McKinlay** – *The 8051 Microcontroller and Embedded Systems*, Pearson
8. **Ytha Yu & Charles Stanley** – *Microprocessor & Microcontroller Fundamentals*, Wiley

MARKU

MCA 3rd Semester Syllabus

MMCA0303-T (Design and Analysis of Algorithms)

Unit	Topic	Proposed Lectures			
I	Introduction to Algorithms and Complexity <ul style="list-style-type: none">• Definition of Algorithm, Characteristics, and Importance• Pseudocode and Implementation• Time and Space Complexity Analysis• Asymptotic Notations: Big-O, Big-Ω, Big-Θ• Recurrences and Methods of Solving Recurrences (Substitution, Recursion Tree, Master Theorem)	12			
	Algorithm Design Techniques – I <ul style="list-style-type: none">• Brute Force Techniques• Divide and Conquer: Merge Sort, Quick Sort, Binary Search• Greedy Algorithms: Activity Selection, Minimum Spanning Tree (Prim's and Kruskal's), Huffman Coding• Backtracking: N-Queens Problem, Hamiltonian Cycle, Subset Sum Problem		12		
	Algorithm Design Techniques – II <ul style="list-style-type: none">• Dynamic Programming: Fibonacci, Matrix Chain Multiplication, Longest Common Subsequence, Knapsack Problem• Branch and Bound: Traveling Salesman Problem, 0/1 Knapsack Problem• Comparison of Greedy, Dynamic Programming, and Backtracking Approaches• Examples and Problem Solving Using Each Technique			12	
	Graph and Advanced Algorithms <ul style="list-style-type: none">• Graph Representation: Adjacency Matrix/List• Graph Traversals: BFS, DFS• Shortest Path Algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall• Minimum Spanning Tree Algorithms: Prim's, Kruskal's• Network Flow and Matching Problems (Introduction)				12
	Algorithm Analysis and Emerging Topics <ul style="list-style-type: none">• Complexity Classes: P, NP, NP-Complete, NP-Hard• Approximation Algorithms• String Matching Algorithms: KMP, Rabin-Karp• Computational Geometry Basics: Convex Hull, Line Segment Intersection• Randomized Algorithms and Probabilistic Analysis• Case Studies and Optimization Problems				

Recommended Textbooks

1. **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein** – *Introduction to Algorithms*, 3rd Edition, MIT Press
2. **Sartaj Sahni** – *Data Structures, Algorithms, and Applications in C++*, Universities Press
3. **Aho, Hopcroft & Ullman** – *Design and Analysis of Algorithms*, Pearson
4. **G. Brassard & P. Bratley** – *Fundamentals of Algorithmics*, Prentice Hall

 **Reference Books**

5. **E. Horowitz, S. Sahni & S. Rajasekaran** – *Fundamentals of Computer Algorithms*, Galgotia Publications
6. **R. Sedgewick & K. Wayne** – *Algorithms*, 4th Edition, Addison-Wesley
7. **Mark Allen Weiss** – *Data Structures and Algorithm Analysis in C++*, Pearson
8. **Jon Kleinberg & Éva Tardos** – *Algorithm Design*, Pearson
9. **Steven Skiena** – *The Algorithm Design Manual*, Springer

MAKJU

MCA 3rd Semester Syllabus

MMCA0304-T(A) (Computer Oriented Numerical analysis)

Unit	Topic	Proposed Lectures
I	Introduction to Numerical Methods <ul style="list-style-type: none"> • Importance and Applications of Numerical Methods in Computing • Error Analysis: Round-Off and Truncation Errors • Significant Digits and Accuracy • Iterative vs. Direct Methods • Floating-Point Representation and Computer Arithmetic 	12
	Solution of Algebraic and Transcendental Equations <ul style="list-style-type: none"> • Bisection Method • Method of False Position (Regula-Falsi) • Newton-Raphson Method • Secant Method • Convergence Criteria and Error Estimation 	12
III	Interpolation and Approximation <ul style="list-style-type: none"> • Finite Differences, Forward and Backward Difference Tables • Newton's Forward and Backward Interpolation • Lagrange Interpolation Formula • Divided Differences and Newton's Divided Difference Formula • Cubic Splines and Least Squares Approximation 	12
	Numerical Differentiation and Integration <ul style="list-style-type: none"> • Numerical Differentiation Formulas • Error Analysis in Numerical Differentiation • Trapezoidal Rule • Simpson's 1/3 and 3/8 Rule • Gaussian Quadrature and Romberg Integration 	12
V	Numerical Solution of Linear and Differential Equations <ul style="list-style-type: none"> • Gauss Elimination Method • Gauss-Seidel Iterative Method • Matrix Inversion by Numerical Methods • Solution of Ordinary Differential Equations (Euler, Modified Euler, Runge-Kutta 2nd & 4th Order) • Numerical Solution of Partial Differential Equations: Finite Difference Method, Boundary Value Problems 	12

Recommended Textbooks

1. **S. S. Sastry** – *Introductory Methods of Numerical Analysis*, Prentice Hall India
2. **K. E. Atkinson** – *An Introduction to Numerical Analysis*, Wiley
3. **M. K. Jain, S. R. K. Iyengar, R. K. Jain** – *Numerical Methods for Scientific and Engineering Computation*, New Age International
4. **C. F. Gerald & P. O. Wheatley** – *Applied Numerical Analysis*, Pearson

 **Reference Books**

5. **E. Balagurusamy** – *Numerical Methods*, McGraw Hill
6. **R. L. Burden & J. D. Faires** – *Numerical Analysis*, Cengage
7. **F. B. Hildebrand** – *Introduction to Numerical Analysis*, McGraw Hill
8. **Mohan C. Joshi** – *Computer Oriented Numerical Methods*, PHI Learning

MARKU

MCA 3rd Semester Syllabus

MMCA0305-T(A)(Information System Design)

Unit	Topic	Proposed Lectures			
I	Introduction to Information Systems (IS) <ul style="list-style-type: none">• Definition, Importance, and Types of Information Systems• Components of IS: Hardware, Software, Data, People, and Processes• Role of IS in Decision Making and Business Process Improvement• System Development Life Cycle (SDLC) Overview• Emerging Trends in IS: Cloud Computing, IoT, and AI Integration	12			
	System Analysis and Requirements Gathering <ul style="list-style-type: none">• Understanding System Analysis: Objectives and Techniques• Requirement Collection Methods: Interviews, Questionnaires, Observation, JAD• Functional vs. Non-Functional Requirements• Feasibility Study: Operational, Technical, Economic, and Schedule• Data Flow Diagrams (DFD) and Process Modeling		12		
	System Design Principles <ul style="list-style-type: none">• Introduction to System Design: Concept and Methodology• Logical vs. Physical Design			12	
	III <ul style="list-style-type: none">• Design of Input, Output, and User Interfaces• Database Design: ER Model, Normalization, Relational Schema• Modular Design, Structured Design Techniques• UML Diagrams: Use Case, Class, Sequence, Activity, State Diagrams				12
	Implementation, Testing, and Maintenance <ul style="list-style-type: none">• Implementation Planning and System Development Approaches• Coding Standards and Documentation				
IV <ul style="list-style-type: none">• Testing Strategies: Unit, Integration, System, Acceptance• Verification and Validation Techniques• System Deployment and User Training• Maintenance: Corrective, Adaptive, Perfective, and Preventive	12				
Advanced Topics and Emerging Trends <ul style="list-style-type: none">• Enterprise Systems and ERP Design Principles• Decision Support Systems (DSS) and Executive Information Systems (EIS)		12			
V <ul style="list-style-type: none">• Web-Based and Mobile IS Design Considerations• Security, Privacy, and Ethical Issues in IS• Performance Metrics and System Evaluation• Case Studies and Best Practices in IS Design			12		

Recommended Textbooks

1. **James A. Senn** – *Analysis and Design of Information Systems*, McGraw Hill
2. **Shelly Cashman Series** – *Systems Analysis and Design*, Cengage Learning
3. **Alan Dennis, Barbara Haley Wixom, Roberta M. Roth** – *Systems Analysis and Design*, Wiley
4. **Kendall & Kendall** – *Systems Analysis and Design*, Pearson

Reference Books

5. **E. M. Awad** – *Systems Analysis and Design*, Pearson
6. **Gary B. Shelly, Thomas J. Cashman** – *Modern Systems Analysis and Design*, Cengage
7. **Hoffer, George, Valacich** – *Modern Systems Analysis and Design*, Pearson
8. **Leon & Leon** – *Introduction to Information Systems*, Vikas Publishing
9. **Jeffrey A. Hoffer** – *Analysis and Design of Information Systems*, Pearson

MCA 3rd Semester Syllabus

MMCA0306-P(VISUAL BASIC PROGRAMMING & ASSEMBLY LANGUAGE LAB)

1. **Simple Calculator using Visual Basic**
 - Design a GUI that performs basic arithmetic operations like addition, subtraction, multiplication, and division using buttons and textboxes.
2. **Student Management System**
 - Create a program to store and manage student information such as name, roll number, and marks. Include features like adding, updating, and deleting student records using arrays or a database.
3. **Simple Banking System**
 - Develop a system that allows users to create accounts, deposit money, withdraw money, and check balances. Use classes to define account properties and methods.
4. **Temperature Converter**
 - Create an application that converts temperatures between Celsius, Fahrenheit, and Kelvin using input from textboxes and displaying results in labels.
5. **Tic-Tac-Toe Game**
 - Build a simple two-player Tic-Tac-Toe game with a GUI using buttons for the game grid and logic to check for a win or draw.
6. **File Handling Application**
 - Develop a program that allows users to create, read, write, and delete text files. Implement file manipulation operations using Visual Basic file handling methods.
7. **Employee Payroll System**
 - Design a payroll system that computes employee salaries based on hours worked, pay rate, tax deductions, and other benefits.
8. **Student Grading System**
 - Create a program to input student marks, calculate the grade based on predefined criteria (e.g., A, B, C), and display the result.
9. **Basic Inventory Management System**
 - Develop an inventory management system that tracks items in stock, allows users to add, update, and delete products, and calculates total value.
10. **Currency Converter**
 - Implement a currency converter application where users can input an amount in one currency and convert it to another based on real-time conversion rates.

Assembly Language:

1. **Hello World Program in Assembly**
 - Write a simple assembly program that outputs "Hello, World!" to the console.
2. **Basic Arithmetic Operations**
 - Implement assembly programs to perform addition, subtraction, multiplication, and division of two numbers input by the user.
3. **Palindrome Checker**
 - Write an assembly program that checks whether a given string is a palindrome.
4. **Factorial Calculation using Recursion**
 - Develop an assembly program that calculates the factorial of a number using recursive calls.
5. **Prime Number Checker**
 - Create an assembly program that checks if a number is prime or not.
6. **String Reversal**
 - Write a program that reverses a string entered by the user and prints the result.
7. **Sorting an Array of Numbers**

- Implement a sorting algorithm (like bubble sort or insertion sort) in assembly language to sort a set of numbers.
- 8. **Count Vowels and Consonants in a String**
 - Write a program that counts and displays the number of vowels and consonants in a string input by the user.
- 9. **Convert Binary to Decimal**
 - Implement an assembly program that converts a binary number to its decimal equivalent.
- 10. **Matrix Multiplication**
 - Create a program that performs matrix multiplication for two matrices entered by the user.

MARKU

MMCA0307-P (COMPUTER GRAPHICS AND ANIMATION LAB)

1. **Line Drawing Algorithms**
 - Implement and compare different line-drawing algorithms such as **DDA (Digital Differential Analyzer)** and **Bresenham's Line Algorithm**. Visualize how each algorithm draws a line between two points.
2. **Circle and Ellipse Drawing Algorithms**
 - Implement algorithms for drawing circles and ellipses, such as **Bresenham's Circle Algorithm** and **Midpoint Circle Algorithm**. Display how these algorithms generate smooth curves.
3. **2D Transformations**
 - Apply basic 2D transformations like **translation, scaling, rotation, and reflection** to geometric shapes. Create an interactive program where users can transform shapes on the screen.
4. **Clipping Algorithms**
 - Implement **Cohen-Sutherland** or **Liang-Barsky** line clipping algorithm to clip lines within a rectangular viewport and handle intersections.
5. **Flood Fill Algorithm**
 - Implement the **Flood Fill Algorithm** to fill a polygon or an area with a color, similar to how the "Paint" tool works in graphic software.
6. **3D Object Rotation and Translation**
 - Implement basic 3D transformations like rotation, scaling, and translation for 3D objects (e.g., cubes, spheres). Visualize the transformation in a 3D coordinate system.
7. **Hidden Surface Removal**
 - Implement a basic hidden surface removal algorithm like **Z-buffer** or **Painter's Algorithm** to determine which surfaces are visible when rendering 3D scenes.
8. **Fractal Generation**
 - Generate fractals like the **Mandelbrot Set** or **Sierpinski Triangle** using recursion and iteration. Display how complex patterns emerge from simple mathematical formulas.
9. **Bezier Curves and Splines**
 - Implement **Bezier Curve** drawing algorithms and demonstrate their use in creating smooth curves for graphical shapes and animations.
10. **Color Models and Color Space Conversion**
 - Implement basic color models like **RGB** and **HSV**, and write programs to convert between different color spaces (RGB to HSV and vice versa).

MCA 4th Semester Syllabus

MMCA0401(A)-T (Software Engineering)

Unit	Topic	Proposed Lectures	
I	Introduction to Software Engineering <ul style="list-style-type: none">• Definition, Nature, and Scope of Software Engineering• Software Process Models: Waterfall, V-Model, Iterative, Spiral, Agile	12	
	<ul style="list-style-type: none">• Software Characteristics, Quality Attributes, and Metrics• Software Development Life Cycle (SDLC) Phases• Software Engineering Standards (ISO, IEEE)• Recent Trends: DevOps, CI/CD, and Agile Practices		
	Requirements Engineering <ul style="list-style-type: none">• Requirement Analysis and Specification• Functional and Non-Functional Requirements		12
	<ul style="list-style-type: none">• Software Requirements Document (SRS)• Requirements Validation and Management• Use Case Modeling and UML Diagrams• Feasibility Study: Technical, Operational, Economic, and Schedule		
	Software Design <ul style="list-style-type: none">• Design Principles and Concepts• Architectural Design: Layered, Client-Server, MVC, Microservices		
III	<ul style="list-style-type: none">• Detailed Design: Modularization, Data Structures, Algorithms• Object-Oriented Design (OOD) and UML Modeling• Design Patterns: Creational, Structural, Behavioral• Case Studies and Best Practices	12	
	Software Testing and Quality Assurance <ul style="list-style-type: none">• Software Testing Fundamentals: Objectives, Levels, Types• Test Case Design Techniques: Black Box, White Box		
IV	<ul style="list-style-type: none">• Unit, Integration, System, and Acceptance Testing• Software Quality Assurance (SQA) Techniques and Metrics• Debugging and Software Reliability• Automated Testing Tools (Selenium, JUnit, TestNG)	12	
	Software Maintenance and Emerging Trends <ul style="list-style-type: none">• Software Maintenance Types: Corrective, Adaptive, Perfective, Preventive• Configuration Management and Version Control (Git, SVN)		
V	<ul style="list-style-type: none">• Software Project Management: Planning, Scheduling, and Risk Management• Software Metrics and Performance Measurement• Emerging Trends: Cloud-Based Software Development, DevSecOps, AI in SE• Case Studies and Industry Applications		

□ Reference Books for Software Engineering:

"Software Engineering: A Practitioner's Approach" by Roger S. Pressman

- A comprehensive and widely used textbook that covers all aspects of software engineering, including testing, quality assurance, and methodologies.

□ "Software Engineering" by Ian Sommerville

- This book covers a range of topics in software engineering, including software development processes, testing, and quality assurance. It's known for its clarity and practical approach.

□ **"Foundations of Software Testing: ISTQB Certification" by Dorothy Graham, Erik van Veenendaal, Isabel Evans**

- A great resource for understanding software testing concepts, with detailed sections on both manual and automated testing methods, aligned with ISTQB certification.

□ **"The Art of Software Testing" by Glenford J. Myers**

- A classic reference on software testing techniques. It covers the fundamental testing strategies and methods, ideal for beginners and advanced learners.

□ **"Software Quality Assurance: From Theory to Implementation" by Daniel Galin**

- This book provides a thorough understanding of software quality assurance, focusing on methodologies, processes, and tools for ensuring software quality.

□ **"Testing Computer Software" by Cem Kaner, Jack Falk, and Hung Q. Nguyen**

- A detailed guide to practical software testing. This book is highly recommended for understanding the real-world practices of software testing.

□ **"Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by Jez Humble and David Farley**

- A reference for understanding the concept of continuous testing and delivery in software engineering, including automation tools and techniques.

MCA 4th Semester Syllabus

MMCA0402(A)-T (DATA WAREHOUSING & Data MINING)

Unit	Topic	Proposed Lectures
I	Introduction to Data Warehousing <ul style="list-style-type: none"> • Data Warehouse Concepts, Need, and Characteristics • Operational vs. Analytical Databases • Data Warehouse Architecture and Components • ETL Process: Extraction, Transformation, and Loading • Data Marts, Metadata, and Data Cube • OLAP vs. OLTP Systems • Multidimensional Data Models 	12
	Data Warehouse Design and Implementation <ul style="list-style-type: none"> • Schema Design: Star, Snowflake, and Fact Constellation Schemas • Fact Tables and Dimension Tables 	
II	<ul style="list-style-type: none"> • Data Preprocessing: Cleaning, Integration, Reduction, Transformation • Data Summarization and Aggregation • Concept Hierarchies and Attribute Selection • OLAP Operations: Roll-up, Drill-down, Slice, Dice, Pivot 	12
III	Introduction to Data Mining <ul style="list-style-type: none"> • Data Mining Functionalities and Applications • KDD Process (Knowledge Discovery in Databases) 	
	<ul style="list-style-type: none"> • Issues and Challenges in Data Mining • Data Mining Techniques: Classification, Clustering, Association Rules, Regression • Data Mining Tools and Trends • Applications in Business Intelligence, Healthcare, and Finance 	12
IV	Data Mining Techniques and Algorithms <ul style="list-style-type: none"> • Association Rule Mining: Apriori, FP-Growth Algorithms • Classification: Decision Tree (ID3, C4.5), Naïve Bayes, k-NN • Clustering: K-Means, Hierarchical Clustering, DBSCAN • Outlier Detection and Anomaly Analysis • Evaluation of Classification and Clustering Results 	12
	Advanced Topics and Emerging Trends <ul style="list-style-type: none"> • Web Mining: Web Content, Structure, and Usage Mining • Text Mining and Natural Language Processing Basics 	
V	<ul style="list-style-type: none"> • Temporal and Spatial Data Mining • Big Data Analytics: Hadoop, Spark Framework Overview • Privacy, Security, and Ethical Issues in Data Mining • Data Visualization and Dashboard Tools 	12

Recommended Textbooks

1. **Jiawei Han, Micheline Kamber & Jian Pei** – *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers
→ The definitive textbook for theoretical and applied data mining.
2. **Ralph Kimball & Margy Ross** – *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, Wiley
→ Comprehensive reference for warehouse design and dimensional modeling.
3. **Ponniiah, Paulraj** – *Data Warehousing Fundamentals for IT Professionals*, Wiley
→ Practical approach for warehouse architecture and implementation.
4. **Alex Berson & Stephen J. Smith** – *Data Warehousing, Data Mining & OLAP*, McGraw Hill Education
→ Balanced coverage of warehousing and mining integration.

 **Reference Books**

5. **Sam Anahory & Dennis Murray** – *Data Warehousing in the Real World*, Pearson
6. **Margaret H. Dunham** – *Data Mining: Introductory and Advanced Topics*, Pearson
7. **Ian Witten & Eibe Frank** – *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann
8. **Michael J. A. Berry & Gordon S. Linoff** – *Data Mining Techniques*, Wiley
9. **C.S. R. Prabhu** – *Data Warehousing: Concepts, Techniques, Products and Applications*, PHI Learning

MARKU

MCA 4th Semester Syllabus

MMCA0403(A)-T(Web design technology)

Unit	Topic	Proposed Lectures
I	Introduction to Multimedia and Web Applications <ul style="list-style-type: none"> • Concept of Multimedia Applications • People-to-People and People-to-System Interactions • Audio and Video Interpersonal Applications • Shared Application Tools (Zoom, MS Teams, WebRTC) • Audio-Video Conferencing and Protocols (SIP, RTP) • Multimedia Electronic Mail, Document Transfer (SMTP, MIME) • Server-Based and Cloud-Based Applications (AWS, Azure, Firebase) 	12
	Networking Requirements for Multimedia Systems <ul style="list-style-type: none"> • Networking Basics and Requirements for Multimedia • Network Features: Bandwidth, Latency, Jitter, QoS 	12
II	<ul style="list-style-type: none"> • Performance of Audio and Video Transmission • Multicasting, Streaming, and Image Transfer Techniques • CDN (Content Delivery Networks) and Edge Computing • IPv6 and Real-Time Multimedia Networking 	12
	Networking Solutions for Multimedia Systems <ul style="list-style-type: none"> • Multimedia over LAN, WAN, Circuit WAN • ATM, Frame Relay, and SMDS as Multimedia Carriers 	12
III	<ul style="list-style-type: none"> • Optical Networks and Broadband Technologies • 5G Networks and IoT-Based Multimedia Delivery • Software Defined Networking (SDN) and Network Virtualization • Multimedia over IP (MoIP), Cloud Networking Solutions 	12
	Web Technology and Media Compression <ul style="list-style-type: none"> • Introduction to Web Technologies (HTTP/3, HTTPS, WebSockets, REST APIs) • Encoding, Digitization, and Compression for Audio, Image, and Video 	12
IV	<ul style="list-style-type: none"> • Image Compression Formats: JPEG, PNG, WebP, AVIF, SVG • Audio Compression Formats: MP3, AAC, FLAC, Opus • Video Compression: MPEG, MP4, H.264, H.265 • Adaptive Streaming (HLS, DASH) and Web Performance Optimization 	12
	Content Preparation, Interactivity, and Web Security <ul style="list-style-type: none"> • Content Preparation: HTML5, CSS3, DHTML, XML, JSON, VRML, WebGL • Internet and Proxy Servers, Search Engines, and SEO • Browsers, Plugins, and Helper Applications 	12
V	<ul style="list-style-type: none"> • Client-Side Interactivity: JavaScript ES6+, TypeScript, React, Angular • Component Technologies: JavaBeans, CORBA, Node.js, Web Components • Web Security: HTTPS, SSL/TLS, Firewalls, Encryption, Watermarks • E-Payment Systems (UPI, PayPal, Blockchain) • Performance Benchmarking Tools (Google Lighthouse, GTmetrix) 	12

Recommended Textbooks

1. **Tay Vaughan** – *Multimedia: Making It Work*, 9th Edition, McGraw Hill Education
→ Core foundation for multimedia principles, formats, and web integration.

2. **Kogent Learning Solutions Inc.** – *Web Technologies: HTML, JavaScript, PHP, Java, JSP, ASP.NET, XML, and Ajax*, Dreamtech Press
→ Step-by-step practical approach to modern web technologies and applications.
3. **Thomas A. Powell** – *Web Design: The Complete Reference*, McGraw Hill
→ Comprehensive resource for web architecture, design, and multimedia integration.
4. **Ramesh Bangia** – *Multimedia and Web Technology*, Firewall Media
→ Core-level text suitable for multimedia fundamentals and web design concepts.
5. **N. Krishnamurthy & S. Jagan** – *Multimedia Communication Systems: Techniques, Standards, and Networks*, PHI Learning
→ Advanced insights into multimedia networking, compression, and protocols.

Reference Books (Advanced / Modern Concepts)

6. **P. J. Deitel, H. M. Deitel, A. Deitel** – *Internet and World Wide Web: How to Program*, Pearson
→ Hands-on learning with HTML5, CSS3, JavaScript, and server-side scripting.
7. **Nigel Chapman & Jenny Chapman** – *Digital Multimedia*, Wiley
→ Covers modern multimedia technologies, compression standards, and authoring tools.
8. **Kurose & Ross** – *Computer Networking: A Top-Down Approach*, Pearson
→ Deep understanding of multimedia transmission and network design for the web.
9. **Robin Nixon** – *Learning PHP, MySQL, JavaScript & CSS: A Step-by-Step Guide to Creating Dynamic Websites*, O'Reilly
→ Ideal for backend integration and dynamic website development.
10. **Ethan Marcotte** – *Responsive Web Design, A Book Apart*
→ Specialized book for adaptive and mobile-first modern web design.

MARKU