Dr. RamManohar Lohia Avadh University, Ayodhya U.P.

Study and Evaluation Scheme

MCA (Master of Computer Applications)
(Effective From Session 2020-21)

Year – I Semester – I

MCA 101 – Programming in C & Data Structures (Theory)

Course Details

- **Paper No.: I**
- **Course Code: MCA 101**
- **Course Title: Programming in C & Data Structures (Theory)**
- **Marks Distribution: Internal = 50, External = 100, Total = 150**

🎯 Course Goal

1. **Introduce programming in C and computational techniques widely used by engineers & scientists.**
2. **Develop problem-solving skills using Data Structures.**

📌 Course Objectives

The course should enable students to:

Gain knowledge in C programming.

Learn to calculate algorithm efficiency.

Understand stack & queue data structures and their applications.

Learn sorting & searching techniques.

Understand tree & graph data structures.

Apply problem-solving using data structures.

📖 Detailed Syllabus

UNIT I – Basics of C Programming

- Introduction: History & Structure of C, Standard I/O, Declarations.
- Data Types & Storage Classes: char, int, short, long, unsigned; auto, register, static, extern.
- Operators & Expressions: Numeric, Relational, Logical, Bitwise, Type conversion, Precedence.
- Conditional Execution: if, switch, nesting, break, default.
- Loops & Iterations: while, do, for, break, continue.
- Modular Programming: Arguments (by value), scope, global vars, separate compilation, linkage.

UNIT II – Arrays, Structures, Pointers & Libraries

- Arrays: Representation, 1D & 2D, insertion, deletion, traversal, sequential search, address calculation.
- Strings handling.
- Structures & Unions: Declaration, usage, assigning.
- Pointers: Arithmetic, function arguments, dynamic memory allocation.
- Preprocessor: Macros, conditional compilation.
- Standard C Library: File handling (fopen, fread…), string functions, math functions (log, sin…).

UNIT III – Data Structures & Linked Lists

- Introduction: Terminology, operations, complexity, time-space trade-off.
- **Stacks**: Array & linked representation, push & pop, applications (infix→postfix, postfix evaluation).
- **Recursion**: Definition, examples, Tower of Hanoi, backtracking, tail recursion.
- **Queues**: Array & linked representation, circular queue, deque, priority queue.
- **Linked Lists**: Singly, doubly, header list, insertion, deletion, traversal, searching, polynomial representation, garbage collection.

UNIT IV – Trees & Graphs

- **Trees**: Terminology, binary trees, representations, traversals, threaded binary trees, Huffman algorithm.
- **BST**: Insertion, deletion, complexity, AVL trees, B-trees.
- **Graphs**: Terminology, representations (sequential, adjacency matrix), traversal, spanning trees, minimum cost spanning tree.

UNIT V – Searching, Sorting & File Structures

- **Searching**: Sequential search, binary search, analysis.
- **Hashing**: Hash tables, functions, collision resolution, implementation.
- **Sorting**: Insertion, bubble, quick, merge, heap, sorting on different keys.
- **File Structures**: Storage media, file organization, sequential files, indexing & hashing, primary/secondary indices, B-trees, B+ trees, comparisons.