MCA110: Advanced Data Structure and Algorithms	
Teaching Scheme Lectures: 3 hrs/Week Tutorials: 1 hr/Week Credits: 4	**Examination Scheme** Class Test -12Marks Teachers Assessment - 6Marks Attendance – 12 Marks End Semester Exam – 70 marks

Prerequisite: -

1. Familiarity with the fundamentals of C or other programming language
2. A solid background in mathematics, including probability, set theory

Course Objectives:

1. Understand various data structures like array, linked list.
2. Implement operations like insertion, deletion and traversing mechanism on various data structures.
3. Implement Linear and Non-Linear data structures.
4. Implement sorting/searching technique.
5. Understand and implement advance data structure using Non-Linear data structure.
6. Determine and analyze the complexity of given algorithms.

UNIT I (10 Hours)

Introduction to Algorithm Design and Data Structures: Abstract data types, Fundamental and derived data types. Representation, Primitive data structures. Algorithm Definition, Analysis of Algorithm, Comparison of Algorithms. Top Down and bottom up Approaches, Complexity- time and space. Structured approach to programming.

Arrays: Representation of Arrays (Single and Multidimensional arrays), Address calculation using column and row major ordering, Operations on Arrays. Application of arrays- Matrix Multiplication, Sparse matrix.

UNIT II (10 Hours)

Stacks: Abstract Data Type, Primitive Stack operations: Push & Pop, Array and Linked Implementation of Stack , Application of stack: Prefix and Postfix Expressions, Evaluation of postfix expression, Recursion, Tower of Hanoi

Queues: Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Array and linked implementation of queues in C, Dequeue and Priority Queue.

UNIT III (10 Hours)

Linked lists: Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Polynomial Representation and Addition, Generalized Linked List

UNIT IV (10 Hours)

Trees: Basic terminology, Binary Trees, Binary Tree Representation: Array Representation and Dynamic Representation, Complete Binary Tree, Algebraic Expressions, Extended Binary Trees, Array and Linked Representation of Binary trees, Tree Traversal algorithms: In order, Preorder and Post order, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm

UNIT V (6 Hours)

Searching : Sequential search, Binary Search, Comparison and Analysis Internal Sorting: Insertion Sort, Selection, Bubble Sort, Quick Sort, Two Way Merge Sort, Heap Sort, Radix Sort, Practical consideration for Internal Sorting.

Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search

Algorithm, AVL trees, Introduction to m-way Search Trees, B Trees & B+ Trees . **Hashing:** Hash Function, Collision Resolution Strategies Storage Management: Garbage Collection and Compaction.
UNIT VI (10 Hours) Graphs: Terminology, Sequential and linked Representations of Graphs: Adjacency Matrices, Adjacency List, Adjacency Multi list, Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshal Algorithm and Dijkstra Algorithm, Introduction to Activity Networks

Text and Reference Books

1. Data Structures and Program Design in C, R.L. Kruse, B.P. Leung and C. L. Tondo, PHI, 2008.
2. Data Structures, Seymour Lipscutz, Mcgraw Hill Publication, 2009
3. Data structures using C, Aaron M.Tenanbaum, Pearson education, 2004.
4. Data structure through C, Yashvant Kanetkar, BPB Publication, 2006.

Course Outcomes:

1. Solving problems and simulate the insertion and deletion by using DS methods.
2. Understanding the concept and recognize the basic terminology used in computer programming.
3. Write, Compile and Debug programs in C language and use different data types for writing the programs.
4. Design programs connecting decision structures, loops and functions
5. Understand the dynamic behavior of memory by the use of pointers
6. Use different data structures and create / manipulate basic data files and developing applications for real world problems.

Head
Department of Computer Applications
Faculty of Computer Applications
Invertis University, Bareilly (UP)

Registrar
Invertis University

Dean Academics
Faculty of Computer Applications
Invertis University, Bareilly (U