

Course Title: **Data Structure and Algorithms**

Credit: **3**

Course No.: **CSIT.121**

Nature of the Course: **Theory+Lab**

Total hours: **48**

Level: **B.Sc.CSIT** Year: **First**

Semester: **Second**

### 1. Course Description

Study of basic data structure vocabulary and the concept of an algorithm.

### 2. Course Objectives

- To introduce, fundamental data structures and problem solving paradigms
- To introduce time complexity analysis of problems.
- To study the representation, implementation & applications of data structures.
- To compare alternative implementations of data structures.
- To choose the appropriate data structure for modeling a given problem.

### 3. Specific Objectives and Contents

Specific Objectives	Contents
<ul style="list-style-type: none"><li>• Outline the classification of data type</li><li>• Give typical examples of data type</li><li>• Explain the relevance of data structures in programming.</li></ul>	<b>Unit 1. Introduction to data structures (3 Hrs.)</b> 1.1 Preliminary data type and Abstract data type 1.2 Data structure, Need and Types of Data Structure 1.3 Comparison between ADT and Data Structure 1.4 Review of Array, Structure and Pointer
<ul style="list-style-type: none"><li>• Define an algorithm</li><li>• Explain an algorithm's performance</li><li>• Describe algorithm analysis</li><li>• Explain the notion used in algorithm analysis</li></ul>	<b>Unit 2: Algorithm analysis (2 Hrs.)</b> 2.1 Algorithm – definition, characteristics 2.2 Algorithm vs. Program 2.3 Space complexity, time complexity 2.3 Asymptotic notations (Big O, Omega $\Omega$ , Big $\Theta$ )
<ul style="list-style-type: none"><li>• Describe an array, its dimensionality and declaration</li><li>• Explain the aim of sorting algorithm</li><li>• Describe the types of sorting</li><li>• Explain the classes of sorting algorithm</li><li>• Choose appropriate searching strategy</li></ul>	<b>Unit 3: Array Data Structure (8 Hrs.)</b> 3.1 Introduction to Arrays - array representation 3.2 Advantages and Drawbacks of Arrays 3.2 sorting algorithms with efficiency: Bubble sort, Selection sort, Insertion sort, Merge sort, Quick Sort, Heap Sort, Radix sort, Bucket Sort, Concept of stable and unstable sorting 3.3 Searching Algorithms: Linear Search, Binary Search
<ul style="list-style-type: none"><li>• Describe a Linked List</li><li>• Explain the operations and implementations of Lists.</li><li>• Understand advantages and limitations of Different types of Linked List</li><li>• Create and use balanced Trees</li></ul>	<b>Unit 4: Linked List (10 Hrs.)</b> 4.1 Introduction to Linked List Data Structure 4.2 Implementation of List – static & dynamic representation, 4.3 Singly Linked List, Circular Linked List, Doubly Linked List, Doubly circular Linked List 4.4 Operations on List: Insertion, Deletion, Searching, Merging

	4.5 Applications of Linked List – polynomial manipulation 4.6 Generalized linked list – concept & representation
<ul style="list-style-type: none"> <li>• Describe the stack data structure</li> <li>• Identify two basic modes of implementing a stack</li> <li>• Outline the applications of stacks in computing</li> </ul>	<b>Unit 5: Stacks (7 Hrs.)</b> 5.1 Introduction: Definition, Stack as ADT 5.2 Operations on stack 5.3 Implementation of Stack: Using Arrays and using Linked List 5.4 Application - infix to postfix & prefix, postfix evaluation, bracket matching, recursion 5.5 Concept of Multiple stacks
<ul style="list-style-type: none"> <li>• Describe a queue data structure</li> <li>• Outline the different applications of queues in computing</li> <li>• Explain the operations on a queue</li> <li>• Understand the different type of queue implementation</li> </ul>	<b>Unit 6: Queues (6 Hrs.)</b> 6.1 Introduction: Definition, Queue as ADT 6.2 Operations on Queue 6.3 Implementation of Queue: Using Arrays and using Linked List 6.4 Applications- Printing, Scheduling etc 6.4 Circular queue, Dequeue, Priority Queues 6.5 Concept of Multiple Queues
<ul style="list-style-type: none"> <li>• Give a basic definition of a binary tree and BST</li> <li>• Perform different tree operations</li> <li>• Evaluate arithmetic expressions by means of tree traversals.</li> <li>• Explain the implementation of AVL search trees.</li> </ul>	<b>Unit 7: Trees (6 Hrs.)</b> 7.1 Concept & Terminologies 7.2 Binary tree, Binary Search Tree 7.3 Implementation of Trees: Static and Dynamic 7.4 Operations on BST – create. Insert, delete, traversals (preorder, inorder, postorder), counting leaf, non-leaf & total nodes 7.5 Balanced Trees: AVL trees and Rotations, Red Black Trees 7.6 Applications: Expression tree
<ul style="list-style-type: none"> <li>• Describe the graph theory its applications</li> <li>• Understand different representations of graph</li> <li>• Explain graph traversal.</li> <li>• Implement MST and shortest Path Algorithm</li> </ul>	<b>Unit 8: Graph (6 Hrs.)</b> 8.1 Concept & terminologies 8.2 Graph Representation 8.3 Traversals – BFS & DFS 8.4 Minimum Spanning Trees: Kruskals Algorithm 8.5 Shortest Path Algorithms: Dijkstra Algorithm

## 6. Recommended Books:

1. Horowitz Sahani, **Fundamentals of Data Structures**, Galgotia Publication
2. Data Structure Using C & C++, Langsam Yedidyah, Augenstein Moshe J., Tennenbaum Aaron M., PHI
3. ISRD Group, **Data Structures using C**, Tata McGraw Hill
4. Nitin Upadhyay, SK, **The Design and Analysis of Algorithm**, Kataria & Sons