# **23AD21SC- DATA STRUCTURES**

|  |  |  |  |
| --- | --- | --- | --- |
| Course Category: | Skill Enhancement course | Credits: | 2 |
| Course Type: | Practical | Lecture-Tutorial-Practical: | 0-1-2 |
| Pre-requisite: | Familiarity with fundamental programming concepts such as variables, data types, operators, control structures (loops, conditionals), functions, and basic I/O operations. | Sessional Evaluation:  Univ. Exam Evaluation:  Total Marks: | 30  70  100 |

|  |  |  |  |
| --- | --- | --- | --- |
| Course Objectives: | Students undergoing this course are expected to learn : | | |
| * The course aims to strengthen the ability of the students to identify and apply the suitable data structure for the given real-world problem. * It enables them to gain knowledge in practical applications of data structures. | | |
| Course Outcomes | Upon successful completion of the course, the students will be able to: | | **Blooms level** |
| CO1 | Understand the role of data structures in organizing and accessing data | L2 |
| CO2 | Design, implement and apply linked lists for dynamic data storage | L3 |
| CO3 | Develop applications using stacks and queues | L5 |
| CO4 | Design and implement algorithms for operations on binary trees and binary search trees | L5 |
| CO5 | Design novel solutions to small scale programming challenges involving data structures such as stacks, queues, Trees | L5 |
| Course Content  Course Content | UNIT-I  **Introduction to Data Structures:** Definition and importance of Data structures, Abstract data types (ADTs) and its specifications.  **Arrays:** Introduction, 1-D, 2-D Arrays, accessing elements of array, Row Major and Column Major storage of Arrays  **Searching Techniques:** Linear & Binary Search  **Sorting Techniques:** Bubble sort, Selection sort, Quick sort.  **Sample Experiments**   1. Program to find min & max element in an array. 2. Program to implement matrix multiplication. 3. Find an element in given list of sorted elements in an array using Binary search. 4. Implement Selection and Quick sort techniques.   UNIT-II  **Linked Lists:** Singly linked lists representation and operations, doubly linked lists and circular linked lists, Comparing arrays and linked lists, Applications of linked lists.  **Sample Experiments**   1. Write a program to implement the following operations. 2. Insert b. Deletion c. Traversal 3. Write a program to store name, roll no, and marks of students in a class using circular double linked list. 4. Write a program to perform addition of given two polynomial expressions using linked list   UNIT-III  **Stacks:** Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, Applications of stacks in expression evaluation, backtracking, reversing list etc.  **Sample Experiments**   1. Implement stack operations using   a. Arrays b. Linked list.   1. Convert given infix expression into post fix expression using stacks. 2. Evaluate given post fix expression using stack. 3. Write a program to reverse given linked list using stack.   UNIT-IV  **Queues:** Introduction to queues: properties and operations, Circular queues, implementing queues using arrays and linked lists,Applications of queues scheduling, etc.  **Deques:** Introduction to deques (double-ended queues), Operations on deques and their applications.  **Sample Experiments**   1. Implement Queue operations using   a. Arrays b. Linked list   1. Implement Circular Queue using   a. Arrays b. Linked list   1. Implement Dequeue using linked list.   UNIT-V  **Trees:** Introduction to Trees, Binary trees and traversals, Binary Search Tree – Insertion, Deletion & Traversal.  **Sample Experiments**   1. Implement binary tree traversals using linked list. 2. Write program to create binary search tree for given list of integers. Perform in-order traversal of the tree. Implement insertion and deletion operations. | | |
| Text Books &  References  Books | **TEXT BOOKS:**   1. Data Structures and algorithm analysis in C, Mark Allen Weiss, Pearson, 2nd Edition. 2. Fundamentals of data structures in C, EllisHorowitz, SartajSahni, Susan AndersonFreed, Silicon Press, 2008.   **REFERENCE BOOKS:**   1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders. 2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft. 3. Problem Solving with Algorithms and Data Structures by Brad Miller and David Ranum. 4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms" by Robert Sedgewick. | | |