

19CS202 DATA STRUCTURES

Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	30	5	5	30	20	5	5



[https://
www.udemy.com/
javascript-
datastructures/](https://www.udemy.com/javascript-datastructures/)

PREREQUISITE COURSES: Programming for Problem Solving I & II.

COURSE DESCRIPTION AND OBJECTIVES:

This course is aimed at offering fundamental concepts of data structures and explains how to implement them. It begins with the basic concepts of data, data structures and then introduces the primitive and non-primitive data structures in detail. It forms the basis for understanding various ways of representing data and its usage in different computing applications.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand the organization of several ADTs and the manipulation (searching, insertion, deletion, traversing) of data stored in various data structures.	1
2	Apply different data structures to solve a given problem.	1
3	Analyze the efficiency of using different data structures and choose the efficient data structure for solving a given problem.	2
4	Develop new algorithms to solve various problems.	3

SKILLS:

- ✓ *Analyze the data structure required for various applications.*
- ✓ *Handling collisions.*
- ✓ *Implement array or linked list for a given problem.*
- ✓ *Describe Pros & Cons of each data structure.*
- ✓ *Usage of trees and graphs.*

UNIT - I**L-9**

DATA STRUCTURES BASICS: Basic terminology - data, information, data type; Data structures-introduction, storage structures - sequential and linked storage representations; Classification of data structures; Applications of data structures.

SORTING: Selection sort, Bubble sort, Insertion sort, Quick sort, Merge sort.

SEARCHING: Linear search and binary search.

UNIT - II**L-9**

LINKED LISTS: Introduction, Types of linked list -singly linked list, doubly linked list and circular linked list, representation of linked list, operations of linked list; Traverse forward/ reverse order, searching, insertion into, deletion from linked lists; Multi lists; Applications of linked lists.

UNIT - III**L-9**

STACKS AND QUEUES: Stacks - introduction, array and linked representations, implementation and their applications; Queues - introduction, array and linked representations, implementation and their applications;Types - linear, circular and doubly ended queues-operations; Applications of queues.

UNIT - IV**L-9**

TREES: Introduction, Properties, Binary tree - introduction, properties, array and linked representations; Tree traversals and their implementation; Expression trees; BST- definition and operations; AVL trees - definition and construction of AVL trees; Applications of binary trees.

UNIT - V**L-9**

GRAPHS: Introduction, Properties, Graphs representations - adjacency matrix, adjacency list, set representation; Traversals - breath first search and depth first search; Applications of graphs.

Hashing: Introduction, Different hash functions, Collision-collision avoidance, handling methods.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS

TOTAL HOURS: 30

1. Write a C program to implement the following sorting methods to arrange a given list of data items (number of data items ≥ 5000) in ascending/descending order:
 - a) Selection Sort.
 - b) Insertion sort.
 - c) Bubble Sort.
 - d) Measure the performance of each of the above sorting technique and compare with their theoretical time complexities.
2. Write a C program to implement the following sorting methods to arrange a given list of data items(number of data items ≥ 5000) in ascending/ descending order:
 - a) Quick sort.
 - b) Merge sort.
 - c) Measure the performance of each of the above sorting technique and compare with their theoretical time complexities.
3. Write a C program to implement the following searching techniques on a given list of data items organized in the form of array:
 - a) Linear search.
 - b) Binary search.
 - c) Analyze the performance of each searching technique and write down your observations.
4. Write a C program to perform the following operations on Singly Linked List:
 - a) Create a Singly linked list.
 - b) Count the number of nodes in SLL.
 - c) Insertion and Deletion operations at Front, at end and at a given position.
 - d) Traversal.
 - e) Search a given element (KEY) in SLL.
5. Write a C program to perform the following operations on Doubly Linked List:
 - a) Create a Doubly linked list.
 - b) Count the number of nodes in DLL.
 - c) Insertion and Deletion operations at Front, at end and at a given position.
 - d) Traversal.
 - e) Search a given element (KEY) in DLL.
6. Write a C program to implement the STACK using an array and linked list and perform. following operations:
 - a. **Push** an Element on to Stack
 - b. **Pop** an Element from Stack
 - c. Demonstrate how Stack can be used to check given string is Palindrome
 - d. Demonstrate **Overflow** and **Underflow** situations on Stack

7. Write a C program to evaluate POSTFIX expression using STACK.
8. Write a C Program to implement Queue by using an array and linked list and perform following operations:
 - a) **Enqueue**: add element to end of queue.
 - b) **Dequeue**: remove element from front of queue.
 - c) **IsEmpty**: check if queue is empty.
 - d) **IsFull**: Check if queue is full.
9. Write a C Program to implement following operations on Binary Search Tree(BST).
 - a) Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2.
 - b) Traverse the BST in In order, Preorder and Post Order.
 - c) Search a given element (KEY) in BST and report the appropriate message.
10. Write C programs for implementing the following graph traversal techniques:
 - a) Depth first search traversal.
 - b) Breadth first search traversal.
 - c) Measure the time required to perform each traversal operation.

TEXT BOOK:

1. D. Samantha, "Classic Data Structures", 2nd edition, Eastern Economic Prentice-hall Private limited Press, 2000.

REFERENCE BOOKS:

1. Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures", Illustrated edition, Computer Science Press, 2006.
2. Mark Allen Weiss, "Algorithms, Data Structures, and Problem Solving with C++ Illustrated", 2nd edition, Addison-Wesley Publishing Company, 2002.
3. R.G. Dromey and Pearson, "How to Solve it by Computer", 2nd edition, Impression Education, 1998.