COURSE TITLE : **DATA STRUCTURES**
COURSE CODE : **4133**
COURSE CATEGORY : **A**
PERIODS/WEEK : **5**
PERIODS/SEMESTER : **75**
CREDITS : **5**

TIME SCHEDULE

MODULE	TOPICS	PERIODS
1	**STACK AND QUEUE**	19
2	**LIST AND LINKED LIST**	19
3	**TREES**	19
4	**GRAPH**	18

Course General Outcomes:

Sl.	G.O	On completion of this course the student will be able :
1	1	Introduction to different Data Structures
	2	Understanding Stack and its operations
	3	Understanding Queues and its operations
2	1	Understanding list and its operations
	2	Understanding linked list and its operations
3	1	To understand non linear data structure - trees
4	1	To understand non linear data structure - graphs
	2	To understand different sorting and searching concepts

Specific outcomes:

MODULE I STACK AND QUEUE

1.1 Introduction to different Data Structures
    1.1.1 Explain efficiency of algorithms, complexity and big O notation.
    1.1.2 Describe different data structures-linear and non linear
    1.1.3 Describe basic data structure operations – insertion, deletion, search, traverse
    1.1.4 Explain about Abstract Data Types (ADTs) and C++ classes
    1.1.5 Explain the use of iterators
    1.1.6 Describe the Array as an ADT with printArray() operation.

1.2 Understanding Stack and its operations
    1.2.1 Describe Stack and its operations - Push and Pop.
    1.2.2 Explain about array representation of stacks
    1.2.3 Describe Stack ADT with push(), pop(), stackfull() and stackempty()

1.2.4 Describe infix, prefix and postfix Expressions

1.2.5 Explain infix to postfix conversion using Stack ADT

1.2.6 Explain evaluation of postfox expression using stack ADT

1.3 Understanding Queues and its operations

1.3.1 Describe Queue and its operations – Insert and Delete.

1.3.2 Describe circular queue and its array representation

1.3.3 Describe Queue ADT (for circular queue) with insert(), delete(), QEmpty() and QFull().

1.3.4 Describe Priority Queue and Dequeue

MODULE II LIST AND LINKED LIST

2.1 Understanding list and its operations

2.1.1 Describe list (using array) and its operations – Find, MakeEmpty, PrintList etc.

2.1.2 Describe List ADT with find(), makeEmpty(), printList(), findKth() etc.

2.2 Understanding linked list and its operations

2.2.1 Explain linked list and its operations – Find, MakeEmpty, PrintList, FindKth, Insert, Delete, Successor, Predecessor etc.

2.2.2 Describe the methods of memory allocation and deallocation for nodes.

2.2.3 Describe LinkedList ADT with find(), makeEmpty(), printList(), findKth(), insert(), delete() etc.

2.2.4 Describe algorithm for implementing stack with LinkedList ADT.

2.2.5 Describe algorithm for implementing queue with LinkedList ADT.

2.2.6 Discribe about doubly linked lists and circular linked lists.

MODULE III TREES

3.1 Understanding Trees and its operations

3.1.1 Explain binary tree, key terms related to trees and traversal methods.

3.1.2 Explain Linked representation of binary trees

3.1.3 Explain binary search trees (BST) and its operations – traversals, insertion, deletion and find.

3.1.4 Describe BST ADT with inOrder(), preOrder(), postOrder(), insert(), delete(), find() etc.

3.1.5 Describe Expression trees and Threaded binary trees.

MODULE IV GRAPH

4.1 Understanding Graph and its operations

4.1.1 Explain graph and key terms related to graphs

4.1.2 Explain graph representations – adjacency matrix mehod and adjacency list method.

4.1.3 Describe graph traversals – DFS and BFS

4.1.4 Describe Graph ADT with dfs() and bfs() methods.

4.1.5 Describe and implement Warshall's algorithm for all-pairs shortest path

4.2 Understanding Searching and Sorting

4.2.1 Explain and implement linear search and binary search algorithms

4.2.2 Explain and implement bubble sort and quick sort algorithms

CONTENT DETAILS

MODULE I : STACK AND QUEUE

Efficiency of algorithm-complexity- Data structures-linear and non linear-insertion, deletion, search, traverse- Iterators- Abstract Data Types (ADTs) – Array as an ADT .

Stack- representation- stack as an ADT - push() and pop() operations -Infix, Prefix, Postfix Expressions- Applications of stack-infix to postfix conversion-postfix evaluation.

Queue-Array representation of circular queue- queue as an ADT - insert() and delete() operations - Priority Queue and Dequeue.

MODULE II : LIST AND LINKED LIST

List using array – List ADT – find(), makeEmpty(), printList() and findKth() operations. Linked List - Memory allocation and de-allocation – LinkedList ADT – find(), makeEmpty(), printList(), findKth(), insert() operations – stack with LinkedList, queue with LinkedList – doubly linked list and circular list

MODULE III : TREES

Binary trees – key terms - Linked representation -binary search trees, BST ADT- inOrder(), preOrder(), postOrder(), insert(), delete(), find() operations – Expression trees and Threaded binary trees

MODULE IV : GRAPH

Graph Definition- Key terms - Representations – Traversal Algorithms – BFS,DFS – Graph ADT – All-pairs Shortest Path Algorithm.

Searching and Sorting Algorithms-Linear search and Binary search - Bubble sort and Quick sort algorithms.

TEXTBOOK

1. Fundamentals of Data Structures in C++, E.Horowitz- S.Sahni, Galgotia-2006
2. Data Structures and Algorithm Analysis in C++, M.A.Weiss, Pearson Education-Fourth Edition

REFERENCES

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, University Press
2. **Data Structures using C and C++ by Yedidyah Langsam, Moshe J. Augenstein and Aron M. Tananbaum, PHI.2002**