DATA STRUCTURES&ALGORITHMS	
CSE 211	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

Basic Knowledge of Programming Fundamentals
Knowledge of Programming Languages (C)

Course Objectives:

The course should enable the students:

- To acquire knowledge on several linear and nonlinear data structures like stacks, queues, linked list, trees and graphs.
- To have better insight into to learn various sorting and searching techniques.
- To exercise the applications of data structures.
- To have a good understanding of problem solving using data structure tools and techniques.

Course Outcomes:

the end of the course, the student will be able to:	
1.	Analyze the complexities of recursive and Non recursive algorithms and Implement linear, binary, interpolation, hashing searching techniques and sorting techniques namely bubble, insertion, selection, quick, merge sort.
2.	Apply ADT concepts such as stacks and queues for solving infix to post fix, postfix evaluation, priority queues.
3.	Apply the concepts of dynamic memory allocation for reducing the time and space complexity of algorithms.
4.	Design and implement the Nonlinear data structures (trees) to optimize the solution.
5.	Design and Implement Warshall’s Algorithm, Shortest path Algorithm-Dijkstra’s Algorithm, Minimum cost spanning trees (Prims and Krushkals algorithms),Graph traversals (Breadth first search and Depth first Search algorithms.)

Mapping of Course Outcomes with Program Outcomes:

S.No	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO12	PSO1	PSO2
CO 1	3	3	2	3	1	1	0	1	1	3	0	2	3	0
CO 2	2	2	3	2	0	0	0	1	1	2	0	2	2	0
CO 3	2	2	3	2	0	0	0	1	1	2	0	2	3	0
CO 4	2	3	3	2	0	0	0	1	1	2	0	3	2	0
CO5	2	3	3	3	0	0	0	1	1	2	0	3	3	0

SYLLABUS

UNIT-I:

15 periods

Introduction: Basic Terminology, Elementary Data Organization, Data Structure operations, Fundamentals of analysis of algorithms and efficiency – Asymptotic Notations and Basic Efficiency classes.

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C, Character string operation, Array as Parameters, Sparse Matrices.

Searching &Sorting: Sequential search, binary search, Interpolation Search, comparison and analysis, Hash Table, Hash Functions. Complexity of Search Algorithm, Insertion Sort, Bubble Sort, Selection Sort, Quick Sort, Merge Sort.

CO1: Analyze the complexity of Algorithms, Implement searching and soring algorithms.
CO2: Implement the searching and soring algorithms.

UNIT-II:

12 periods

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Applications of stack: Conversion of Infix to prefix and Postfix Expressions, Evaluation of Postfix & Prefix expressions using stack, Recursion, Towers of Hanoi Problem.

Queues: Array representation and implementation of queues, Operations on Queue: Insert, Delete, Full and Empty. Circular queue, De-queue, and Priority Queue, Applications of Queues.

CO1: Implement stacks and queues using ADT and Implement the applications of Stacks and queues (solving infix to post fix, postfix evaluation, priority queues.)

CO2: Apply ADT and implement Stack and queue and applications of stack and queue.

UNIT-III:

12 periods

Linked list: Representation and Implementation of Singly Linked Lists, Traversing and Searching of Linked List, Insertion and deletion to/from Linked Lists, Doubly linked list, Circular doubly linked list, Implementing priority queue using Linked List, Polynomial Representation using Linked list & addition.

CO1: Implement singly linked list, Doubly Linked List, Circular doubly linked list and applications.

CO2: Implement Linked Lists and applications of Linked Lists.

UNIT-IV:

12 periods

Trees: Basic terminology, Binary Trees, Binary tree representation, Almost Complete Binary Tree, Complete Binary Tree, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees. Binary Search Tree (BST), Insertion and Deletion in BST, AVL Trees-Rotations in AVL trees, Insertion and Deletion in AVL.

CO1: Design and implement BST, AVL trees.

CO2: Implement BST, AVL tree along with various operations performed on BST and AVL tree.

UNIT-V:

12 periods

Graphs: Terminology & Representations- Graphs, Directed Graphs, Adjacency Matrices, Path OR Transitive Closure of a Graph, Warshall's Algorithm, Shortest path Algorithm-Dijkstra's Algorithm, Connected Component and Spanning Trees, Minimum Cost Spanning Trees, Graph Traversals.

CO1: Implement Graph Traversals algorithm and Minimum Cost Spanning Trees algorithms.

CO2: Implemnet Warshall's Algorithm, Shortest path Algorithm-Dijkstra's Algorithm and Minimum Cost Spanning Trees algorithm

TEXT BOOKS

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education, 2nd Edition, 1996

REFERENCE BOOKS

1. E.Horowitz and Sahani, "Fundamentals of Data Structures", W H Freeman & Co Publication, 1983.
2. S. Lipschutz, "Data Structures", McGraw Hill Publications, 1986.
3. P. Dey & M. Ghosh, "Programming in C" , Oxford Univ. Press, 2012
4. ISRD Group, "Data Structures through C++", McGraw Hill, 2011.

Web Resources:

1. https://nptel.ac.in/courses/106/102/106102064/
2. https://www.edx.org/course/foundations-of-data-structures
3. https://www.pluralsight.com/courses/ads-part1?aid=7010a000002BWq6AAG&promo=&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=IN_Dynamic&utm_content=&gclid=CjwKCAjwh472BRAGEiwAvHVfGkBPYPTtMyZXFPv0dFT447PrKa_n8BKqox2DhR-zBq7s4EvOubgD0hoCUyIQAvD_BwE

Prepared By: N. Lokeswari, K. Amaravathi