DATA STRUCTURES AND ALGORITHMS

SUBJECT CODE: CE 304

Teaching Scheme (Credits and Hours)

Teaching scheme				Total Credit	Evaluation Scheme		Mid Sem Exam	CIA	Pract.	Total
L	T	P	Total		Theory					
Hrs	Hrs	Hrs	Hrs		Hrs	Marks	Marks	Marks	Marks	Marks
03	00	02	05	04	03	70	30	20	30	150

Learning Objectives

The objective of the course is to introduce the fundamentals of Data Structures, Abstract concepts and how these concepts are useful in problem solving. After completion of this course student will be able to -
• Analyze step by step and develop algorithms to solve real world problems.
• Implementing various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs.
• Understanding various searching & sorting techniques

Outline of the Course

Sr. No	Title of the Unit	Minimum Hours
1	BASICS	02
2	INTRODUCTION TO DATA STRUCTURE	04
3	LINEAR DATA STRUCTURE	15
4	NONLINEAR DATA STRUCTURE	10
5	SORTING and SEARCHING	06
6	HASHING AND FILE STRUCTURES	08

Total hours (Theory): 45

Total hours (Lab): 30

Total hours: 75

Detailed Syllabus

Sr. No	Topic	Lecture Hours	Weight age(%)
1	**Basics** • Algorithm Specifications: Performance Analysis and Measurement (Time and space analysis of algorithms- Average, best and worst case analysis).	02	5
2	**Introduction To Data Structure:** • Data Management concepts, • Data types – primitive and non-primitive, • Types of Data Structures- Linear & Non Linear Data Structures.	04	5
3	**Linear Data Structure** • Array: Representation of arrays, Applications of arrays, sparse matrix and its representation., • Stack: Stack-Definitions & Concepts, Operations On Stacks, Applications of Stacks, Polish Expression, Reverse Polish Expression And Their Compilation, Recursion, Tower of Hanoi, • Queue: Representation Of Queue, Operations On Queue, Circular Queue, Priority Queue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue, • Linked List: Singly Linked List, Doubly Linked list, Circular linked list ,Linked implementation of Stack, Linked implementation of Queue, Applications of linked list.	15	25
4	**Nonlinear Data Structure** : • Tree-Definitions and Concepts, Representation of binary tree, Binary tree traversal (Inorder, postorder, preorder), • Threaded binary tree, • Binary search trees, • Conversion of General Trees To Binary Trees, • Applications Of Trees- Some balanced tree mechanism, eg. AVL trees, 2-3 trees, Height Balanced, Weight Balance , • Graph-Matrix Representation Of Graphs, Elementary Graph operations,(Breadth First Search, Depth First Search, Spanning Trees, Shortest path, Minimal spanning tree).	10	25
5	**SORTING And SEARCHING** • Insertion Sort, • Quick Sort, • Merge Sort, • Heap Sort, • Sorting On Several Keys, • List and Table Sort,	06	20

	• Linear Search, • Binary Search.		
6	**Hashing And File Structures :** • Hashing: The symbol table, • Hashing Functions, • Collision-Resolution Techniques, • File Structure: Concepts of fields, records and files, • Sequential, Indexed and Relative/Random File Organization, • Indexing structure for index files, • hashing for direct files, • Multi-Key file organization and access methods.	08	20
	TOTAL	**45**	**100**

Instructional method and Pedagogy

- At the start of course, the course delivery pattern, prerequisite of the subject will be discussed.
- Lectures will be conducted with the aid of multi-media projector, black board, OHP etc.
- Attendance is compulsory in lecture which carries 10 marks in overall evaluation.
- One internal exam will be conducted as a part of internal theory evaluation.
- Assignments based on the course content will be given to the students for each unit and will be evaluated at regular interval evaluation.
- Surprise tests/Quizzes/Seminar/tutorial will be conducted having a share of five marks in the overall internal evaluation.
- The course includes a laboratory,where students have an opportunity to build an appreciation for the concepts being taught in lectures.
- Experiments shall be performed in the laboratory related to course contents.

Learning Outcome

On successful completion of the course, the student will:

- Be able to check the correctness of algorithms using inductive proofs and loop invariants.
- Be able to compare functions using asymptotic analysis and describe the relative merits of worst-, average-, and best-case analysis.
- Be able to solve recurrences using the master, the iteration, and the substitution method.
- Become familiar with a variety of sorting algorithms and their performance characteristics (eg, running time, stability, space usage) and be able to choose the best one under a variety of requirements.
- Be able to understand and identify the performance characteristics of fundamental algorithms and data structures and be able to trace their operations for problems such as sorting, searching, selection, operations on numbers, polynomials and matrices, and graphs.

- Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
- Be able to use the design techniques introduced i.e. dynamic programming, greedy algorithm etc. to design algorithms for more complex problems and analyze their performance.
- Become familiar with the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

Reference Books:
1. An Introduction to Data Structures with Applications. by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill.
2. Data Structures using C & C++ -By Ten Baum Publisher – Prenctice-Hall International.
3. Fundamentals of Computer Algorithms by Horowitz, Sahni,Galgotia Pub. 2001 ed.
4. Fundamentals of Data Structures in C++-By Sartaj Sahani.
5. Data Structures: A Pseudo-code approach with C -By Gilberg & Forouzan Publisher-Thomson Learning.

List of Experiments

Sr. No	Name of Experiment
1	**Introduction to structures & pointers in C**.
2	**Stack operations** Write a program to perform PUSH, POP, PEEP & CHANGE operations on Stack.
3	**Queue Operations** Write a program to implement insertion & deletion in a queue.
4	**Circular Queue Operations** Write a program to implement insertion & deletion in a circular queue
5	Write a program for linked list insertion, deletion & copy
6	**Sorting and searching :** Write a program to perform • Selection sort • To sort the given number using bubble sort • Merge sort • Quick sort • Sequential and binary search