Data Structures using C

Subject Code : 18CSI301
Credits : 03

Total Contact Hours: 45
L-T-P: 3-0-0

Prerequisite: Knowledge on Basic Programming using C and Problem Solving Skills.

Course Objectives:

- Explain fundamentals of data structures and their applications essential for programming/problem solving
- Analyze Linear Data Structures: Stack, Queues, Lists
- Analyze Non-Linear Data Structures: Trees, Graphs
- Analyze and Evaluate the sorting & searching algorithms
- Assess appropriate data structure during program development/Problem Solving

Unit I:

(9 Hours)

Introduction: Data Structures, Classifications (Primitive &Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. **Pointers and Dynamic Memory Allocation Functions:** Representation of Linear Arrays in Memory, Dynamically allocated arrays, Multidimensional Arrays, Polynomials and Sparse Matrices. Strings: Basic Terminology, Storing, Operations and Pattern Matching algorithms. **Sorting and Searching**: Insertion Sort, Radix sort, Address Calculation C Programming Examples Sort.

Unit II:

(10 hours)

Stacks and Queues Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, **Stack Applications**: Polish notation, Infix to postfix conversion, evaluation of postfix expression, **Recursion** - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues**: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. C Programming.

Unit III:

(10 hours)

Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. **Hashing**: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. C Programming.

Unit IV:

(8 hours)

Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, **Binary Tree Traversals** - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, **Application of Trees**-Evaluation of Expression, C Programming.

Unit V:

(8 hours)

Graphs: Definitions, Terminologies, Types of Graphs, Matrix and Adjacency List Representation Of Graphs, Elementary Graph operations. Minimal Spanning Tree: Prim's algorithm, Kruskal's Algorithm.**Traversal methods**: Breadth First Search and Depth First Search. Applications of Graph. **Files and Their Organization**: Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing.

Course Outcomes:

At the end of the course, students will be able to:

- Acquire knowledge of
  - Various types of data structures, operations and algorithms.
  - Sorting and searching operations.
  - File structures.
- .
- Analyze the performance of – Stack, Queue, Lists, Trees, Graphs, Searching and Sorting techniques.
- Implement all the applications of Data structures in a high-level language.

Design and apply appropriate data structures for solving computing problems

Text Books:

1. Weiss, Data Structures and Algorithm Analysis in C, IV Edition, Pearson Education, 2014
2. Lipschutz: Schaum's outline series Data structures Tata McGraw-Hill

Reference Books:

1. Kamthane: Introduction to Data Structures in C. Pearson Education 2005.
2. Hanumanthappa M., Practical approach to Data Structures, Laxmi Publications, Fire Wall media 2006
3. Langsam, AusensteinMaoshe& M.Tanenbaum Aaron Data Structures using C and C++ Pearson Education.
4. Robert Kruse Data Structures and program designing using 'C', Trembley and Sorenson Data Structures

Data Structures using C Lab

Subject Code : 18CSI301L
Credits : 01

L-T-P:0-0-2

List of Experiments:

1. **Demonstrating Pointers Usage**
   a) Printing Memory Addresses: Write C program to demonstrate the use of pointers by printing memory address 2.
   b) Writing a Swap Function: Write a C program to swap two numbers using pointers concept
   c) Allocating and Freeing Memory: Write a C program to demonstrate the use of allocating a memory and freeing
   d) Memory Leaks and Other Problems: Write a C program to demonstrate the memory leaks when pointers are not used properly.

2. **Demonstrate Strings, User defined data types and Files in C**
   a) Reading and Writing Strings: Write a C program to demonstrate the input and output operations on strings
   b) String operations / Manipulations: Write a C program to demonstrate the operations on strings – by writing user defined string functions.
   c) Enumerations, Structures and Union: Write a C program to demonstrate Enumerations, Structures and Union data types. Write a program for following using recursive methods.
   d) File operations: Write a C program to demonstrate the input and output operations on files

3. **Demonstrate the technique of recursion in C**
   a) Recursion – Write recursive function for i) Sum of natural numbers ii) Factorial of a given number iii) Fibonacci sequence

4. **Stack ADT** Implement Stack using Arrays

5. **Queue ADT** Implement Queue using Arrays

6. **Singly Linked List** Write a C Program to perform following operations on Singly Linked List ADT: i. Create ii. Insert iii. Delete iv. Display

7. **Doubly Linked List** Write a C Program to perform following operations on Doubly Linked List ADT: i. Create ii. Insert iii. Delete iv. Display

8. **Circular Linked List** Write a C Program to perform following operations on Circular Linked List ADT: i. Create ii. Insert iii. Delete iv. Display

9. Implement Stack using List

10. Implement Queue using List

11. Implement Binary Search Tree – using List

12. i)Implement a simple heap ii) Implement Priority Queue using heap