

## DATA STRUCTURES LABORATORY

[As per Choice Based Credit System (CBCS) scheme]  
(Effective from the academic year 2015 -2016)

### SEMESTER - III

Laboratory Code	15CSL38	IA Marks	20
Number of Lecture Hours/Week	01I + 02P	Exam Marks	80
Total Number of Lecture Hours	40	Exam Hours	03

### CREDITS - 02

#### Course objectives:

This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of

- Asymptotic performance of algorithms.
- Linear data structures and their applications such as stacks, queues and lists
- Non-Linear data structures and their applications such as trees and graphs
- Sorting and searching algorithms

#### Descriptions (if any)

Implement all the experiments in C Language under Linux / Windows environment.

#### Laboratory Experiments:

1. Design, Develop and Implement a menu driven Program in C for the following Array operations
  - a. Creating an Array of N Integer Elements
  - b. Display of Array Elements with Suitable Headings
  - c. Inserting an Element (ELEM) at a given valid Position (POS)
  - d. Deleting an Element at a given valid Position(POS)
  - e. Exit.

Support the program with functions for each of the above operations.

2. Design, Develop and Implement a Program in C for the following operationson Strings
  - a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)
  - b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR

Support the program with functions for each of the above operations. Don't use Built-in functions.

3. Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)
  - a. *Push* an Element on to Stack
  - b. *Pop* an Element from Stack
  - c. Demonstrate how Stack can be used to check *Palindrome*
  - d. Demonstrate *Overflow* and *Underflow* situations on Stack
  - e. Display the status of Stack



f. Exit

Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, \*, /, %(Remainder), ^(Power) and alphanumeric operands.
5. Design, Develop and Implement a Program in C for the following Stack Applications
  - a. Evaluation of Suffix expression with single digit operands and operators: +, -, \*, /, %, ^
  - b. Solving Tower of Hanoi problem with n disks
6. Design, Develop and Implement a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX)
  - a. Insert an Element on to Circular QUEUE
  - b. Delete an Element from Circular QUEUE
  - c. Demonstrate *Overflow* and *Underflow* situations on Circular QUEUE
  - d. Display the status of Circular QUEUE
  - e. Exit

Support the program with appropriate functions for each of the above operations

**Continued:**

7. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: *USN, Name, Branch, Sem, PhNo*
  - a. Create a SLL of N Students Data by using *front insertion*.
  - b. Display the status of SLL and count the number of nodes in it
  - c. Perform Insertion / Deletion at End of SLL
  - d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack)
  - e. Exit
8. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: *SSN, Name, Dept, Designation, Sal, PhNo*
  - a. Create a DLL of N Employees Data by using *end insertion*.
  - b. Display the status of DLL and count the number of nodes in it
  - c. Perform Insertion and Deletion at End of DLL
  - d. Perform Insertion and Deletion at Front of DLL
  - e. Demonstrate how this DLL can be used as Double Ended Queue
  - f. Exit



9. Design, Develop and Implement a Program in C for the following operations on **Singly Circular Linked List (SCLL)** with header nodes
  - a. Represent and Evaluate a Polynomial  $P(x,y,z) = 6x^2y^2z - 4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3$
  - b. Find the sum of two polynomials **POLY1(x,y,z)** and **POLY2(x,y,z)** and store the result in **POLYSUM(x,y,z)**
 Support the program with appropriate functions for each of the above operations
10. Design, Develop and Implement a menu driven Program in C for the following operations on **Binary Search Tree (BST)** of Integers
  - a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
  - b. Traverse the BST in Inorder, Preorder and Post Order
  - c. Search the BST for a given element (**KEY**) and report the appropriate message
  - e. Exit
11. Design, Develop and Implement a Program in C for the following operations on **Graph(G)** of Cities
  - a. Create a Graph of N cities using Adjacency Matrix.
  - b. Print all the nodes **reachable** from a given starting node in a digraph using DFS/BFS method
12. Given a File of N employee records with a set K of Keys(4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table(HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function  $H: K \rightarrow L$  as  $H(K) = K \bmod m$  (**remainder method**), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using **linear probing**.

**Course outcomes:**

On the completion of this laboratory course, the students will be able to:

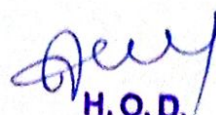
- Analyze and Compare various linear and non-linear data structures
- Code, debug and demonstrate the working nature of different types of data structures and their applications
- Implement, analyze and evaluate the searching and sorting algorithms
- Choose the appropriate data structure for solving real world problems

**Graduate Attributes (as per NBA)**

1. Engineering Knowledge
2. Problem Analysis
3. Design/Development of Solutions
4. Modern Tool Usage

**Conduction of Practical Examination:**

1. All laboratory experiments (TWELVE nos) are to be included for practical examination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script
4. Marks distribution: Procedure + Conduction + Viva: 20 + 50 + 10 (80)
5. Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.

  
H. O. D.