****

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

**An Autonomous Institute Affiliated to VTU, Belagavi Approved by AICTE; ISO 9001:2015**

Certified Accredited by National Assessment Accreditation Council (NAAC) with ‘A’ grade

Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COURSE HANDOUT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Academic Year** | **2023-2024** | **L: T: P: S:** | **3:0:1:0** |
| **Course Title** | |  | | --- | | **Data Structures & its applications** | | **Course code** | **22CS33** |
| |  | | --- | | **Credit Units** | | **04** | **Course Type** | **Integrated** |
| **Course Authors** | **Prof. A M Prasad & Prof. Aruna S** | **Date Approved** | **25/09/2023** |
|  | | | |
| **Exam Hours** | **03** | **CIE Marks** | **50** |
| **Total Hours:** | **52 [T:Theory 40 + L:Laboratory 12 ]** | **SEE Marks** | **50** |

**Integrated Laboratory Pre-Requisites:**

|  |  |
| --- | --- |
| 1. | Certification: Pointers In C Programming |
| 2. | Course Time: 01 hr 45 mins, Advanced level. Student can earn a certificate on completing the content <https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01384203240484864010470_shared/overview> |

**Integrated Laboratory Objectives and Outcomes:**

**Objectives:**

1. Introduce the concept of data structures through ADT including List, Stack, Queues
2. To design and implement various data structure algorithms.
3. To introduce various techniques for representation of the data in the real world
4. To develop application using data structure algorithms

**Outcomes: At the end of the course, student will be able to:**

|  |  |
| --- | --- |
| CO1 | Select appropriate data structures to be used in the programming solution for a given problem. |
| CO2 | Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures. |
| CO3 | Implement Linear and Non-Linear data structures efficiently. |
| CO4 | Implement various operations on Linked Lists |
| CO5 | Design and Implement applications of Non-linear data structure. |
| CO6 | Implement graph traversal algorithms |

**The student should be able to:**

|  |  |
| --- | --- |
| 1 | Analyze and Compare various linear and non-linear data structures |
| 2 | Code, debug and demonstrate the working nature of different types of data structures and their applications |
| 3 | Implement, analyze and evaluate the searching and sorting algorithms |
| 4 | Choose the appropriate data structure for solving real world problems |

**Laboratory exercises**

|  |  |
| --- | --- |
| **Q.no** | **Problem Statement** |
| 1. | 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 |
| 2 | 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. |
| 3 | Design, Develop and Implement a Program in C for the following Stack Applications. Evaluation of Suffix expression with single digit operands and operators: +, -, \*, /, %, ^ |
| 4. | 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 |
| 5. | Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List of Student Data with the fields: USN, Name, Programme, 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 e. Exit |
| 6. | 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 |
| 7. | Design, Develop and Implement Program in C for the following operations on Singly Linked List (SLL) with header nodes  a. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z - 4yz5+3x3yz+2xy5z- 2xyz3  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 |
| 8. | 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 b. Traverse the BST in Inorder, Preorder and Post Order  c. Search the BST for a given element (KEY) and report the appropriate message d. Exit |
| 9. | Design, Develop and Implement a Program in C for the operations on Graph(G) of Cities  a. Create a Graph of N cities using Adjacency Matrix.  b. Print all nodes reachable from a given source node in a graph using DFS/BFS method |
| 10. | 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→L as H (K) =K mod m, and implement hashing technique to map a given key K to the address space L. Resolve the collision if any using linear probing. |