**Course code: CSL 201 Course Name : Data Structures Lab**

**Faculty In Charge: Dr Binu V P**

L-T-P-Credits 0-0-3-2

Pre-requisite: CST 201 Data Structures, EST 102 C programming skills.

Operating System to Use in Lab : Linux

Compiler/Software to Use in Lab : gcc

Programming Language to Use in Lab : Ansi C

Preamble: The aim of the Course is to give hands-on experience for Learners on creating and using different Data Structures. Data Structures are used to process data and arrange data in different formats for many applications. The most commonly performed operations on data structures are traversing, searching, inserting, deleting and few special operations like merging and sorting.

**Lab Cycle-6**

**Learning Outcome:** Implement doubly linked and circular linked list

**Date of submission:** on or before 28-11-2022

**(Write these programs in fair record-show the output in lab and get it signed by the staff in charge. There will be viva voce in every lab. )**

**Doubly-linked list**

A more sophisticated kind of linked list is a doubly-linked list or two-way linked list. Each node has two links: one points to the previous node, or points to a null value or empty list if it is the first node; and one points to the next, or points to a null value or empty list if it is the final node.

**Write programs to implement the following using doubly linked list:**

Create a doubly linked list with n elements by adding elements at the end.

Given a node data, insert a new node after it.

Given a node data, insert a new node before it.

Insert a new node in the given position.

Delete a node, given the key data value.

Delete a node given the position.

Delete the smallest element from the list.

**Circularly-linked list**

In a circularly-linked list, the first and final nodes are linked together. This can be done for both singly and doubly linked lists. To traverse a circular linked list, you begin at any node and follow the list in either direction until you return to the original node. Viewed another way, circularly-linked lists can be seen as having no beginning or end. This type of list is most useful for managing buffers for data ingest, and in cases where you have one object in a list and wish to iterate through all other objects in the list in no particular order.

The pointer pointing to the whole list may be called the access pointer.

**Implement a circular linked list with provisions to insert and delete elements(singly/doubly)**

**Represent polynomials using linked list and write programs to do the**

addition

subtraction.

multiplication.