# Department of Computing

**CS-250: Data Structure and Algorithms**

**Class: BEEE 13 (Grp 1+2)**

# 

# Lab 5: Use knowledge of linked lists to implement a small functionality

**Date: 4rd March, 2024**

**Time: 10 am - 1 pm**

# Lab Instructor: Anum Asif

# 

# Lab 5: Use knowledge of linked lists to implement a small functionality

**Introduction**

Doubly circular linked lists are important data structures which are used in many applications. Students have learned the fundamental concepts of circular linked lists in the lectures. This lab will introduce students with the practical implementation of a doubly circular linked list and different operations that can be performed on it.

**Objectives**

Objective of this lab is to familiarize students with the doubly circular linked list and implement them in C++.

**Tools/Software Requirement**

Visual Studio C++

**Helping Material**

Lecture slides. Text book.

**Description**

This lab is about the implementation of the doubly linked list and circular linked list. In doubly linked list you have two pointers in the link part of node i.e. next and previous pointers. With the help of these pointers it becomes very easy to transverse the list. In circular linked list the successor of tail pointer points to the head pointer or start of the linked list. And every new node is added after the last node but before the first node. Keeping these concepts in mind, solve the lab tasks.

## 

## The basic operation consist of

* ***Creating*** the list.
* ***Inserting*** nodes at the beginning, at the end and at a a specific location.
* ***Deletion*** nodes at a specific location.
* ***Traversing/displaying*** the data in the list.

**Lab Tasks**

Use the partially written C++ code to implement a doubly circular linked list. The following are the important features (to be implemented):

1. Create a doubly circular linked list.
2. Create a function which can insert data at the beginning of the list.
3. Create a function which can insert data at the end of the list.
4. Create a function which can insert data at any location within the list.
5. Create a function which can delete a node at a specific location.
6. Create a function that can display the contents of the linked list.

**Important Note:** Practice your knowledge of OOP with C++ when creating a solution.

**Partial Code:**

/\* C++ Program to Implement Circular Doubly Linked List

\*/

**#include**<iostream>

**#include**<cstdio>

**#include**<cstdlib>

**using** **namespace** std;

/\*

\* Class Declaration

\*/

**class** DoubleCircularList

{

**public**:

**int** counter = 0;

/\*

\* Node Declaration

\*/

**struct** node{

**int** info;

**struct** node \*next;

**struct** node \*prev;

}\*start, \*last;

**DoubleCircularList**(){

start = NULL;

last = NULL;

}

/\*

\*CREATE NODE AND ALLOCATE MEMORY DYNAMICALLY

\*/

node\* **createNode**(**int** value){

**struct** node \*temp;

//your code here

cout << "+++++++++++++ You have to write code for createNode()";

**return** temp;

}

/\*

\*INSERTS ELEMENT AT BEGINNING

\*/

**void** **insertAtStart**(){

**int** value;

//cout<<endl<<"Enter the element to be inserted: ";

//cin>>value;

//your code here

cout << "+++++++++++ You have to write code for insertAtStart()";

}

/\*

\*INSERTS ELEMNET AT LAST

\*/

**void** **insertLast**(){

**int** value;

//cout<<endl<<"Enter the element to be inserted: ";

//cin>>value;

//your code here

cout << +++++++++++ You have to write code for insertLast()";

}

/\*

\*INSERTS ELEMENT AT POSITION

\*/

**void** **insertAtPos**(){

**int** value, pos, i;

cout<<**endl**<<"Enter the element to be inserted: ";

cin>>value;

cout<<**endl**<<"Enter the postion of element inserted: ";

cin>>pos;

//your code here

cout << +++++++++++ You have to write code for insertAtPos()";

}

/\*

\* Delete Node at Particular Position

\*/

**void** **deleteAtPos**()

{

**int** pos, i;

node \*ptr, \*s;

**if** (start == last && start == NULL)

{

cout<<"List is empty, nothing to delete"<<**endl**;

**return**;

}

cout<<**endl**<<"Enter the postion of element to be deleted: ";

cin>>pos;

//your code here

cout << +++++++++++ You have to write code for deleteAtPos()";

}

/\*

\* Display Elements of the List

\*/

**void** **display**(){

//your code here

cout << +++++++++++ You have to write code for display()";

}

};

/\*

\* Main: Contains Menu

\*/

**int** **main**(){

**int** userChoice;

DoubleCircularList cdl;

**while** (1){

cout<<"\n-------------------------------"<<**endl**;

cout<<"Operations on Doubly Circular linked list"<<**endl**;

cout<<"\n-------------------------------"<<**endl**;

cout<<"1.Insert at Beginning"<<**endl**;

cout<<"2.Insert at Last"<<**endl**;

cout<<"3.Insert at Specific Position"<<**endl**;

cout<<"4.Delete at Specific Position"<<**endl**;

cout<<"5.Display List"<<**endl**;

cout<<"6.Exit"<<**endl**;

cout<<"Enter your choice : ";

cin>>userChoice;

**switch**(userChoice){

**case** 1:

cdl.insertAtStart();

**break**;

**case** 2:

cdl.insertLast();

**break**;

**case** 3:

cdl.insertAtPos();

**break**;

**case** 4:

cdl.deleteAtPos();

**break**;

**case** 5:

cdl.display();

**break**;

**case** 6:

**exit**(1);

**default**:

cout<<"Wrong choice"<<**endl**;

}

}

**return** 0;

}

**Solution:**

|  |
| --- |
| Solution |
| Task 1 Code:  Task 1 Output Screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS.