# University of Central Punjab Faculty of

# Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |
| --- | --- |
| **Graded Lab 03** |  |
| * Abstract Classes * Linked List * Doubly LinkedList |  |

**Instructions:**

* Indent your code.
* Comment your code.
* Use meaningful variable names.
* Plan your code carefully on a piece of paper before you implement it.
* Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

# void main() is not allowed. Use int main()

* **You have to work in multiple files. i.e separate .h and .cpp files**

# You are not allowed to use system("pause")

* **You are not allowed to use any built-in functions**

# You are required to follow the naming conventions as follow:

* + **Variables:** firstName; (no underscores allowed)
  + **Function:** getName(); (no underscores allowed)
  + **ClassName:** BankAccount (no underscores allowed)

# Students are required to complete the following tasks in lab timings.

|  |  |
| --- | --- |
| **Marks Division for Graded Task** | |
| **Submission:** | **Mark(s) = 1** |
| **Using proper naming conventions (functions, variables):** | **Marks = 1** |
| **Proper filing(cpp, header):** | **Marks = 2** |
| **LinkedList Implementation:** | **Mark(s) = 2** |
| **Implementation of ‘*RotateAndDelete’* function:** | **Marks = 4** |
| **Total:** | **10 Marks** |

struct Node

{

int data;

Node\* next;

Node\* prev;

};

Create a C++ generic abstract class named as **LinkedList**, with the following:

**Attributes:**

1. Node \*head
2. Node \*tail

**Functions:**

virtual void insertAtTail(int) = 0;

* + Should add element at the tail node of linkedList.

virtual void deleteAtTail() = 0;

* + Should delete element at the tail node of linkedList.

LinkedList();

Using the above ‘**LinkedList** class, make another derived class named as ‘**DoublyLinkedList’** with the following functionalities:

**void** [**display()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : displays the linked list.

**Graded Task**

* *Now create a function named ‘****RotateAndDelete’.***
* *The function should take two integers as input ‘****m’*** *and ‘****n’****, then move the* ***Nth*** *node to the position* ***‘m’,*** *and then delete the* ***Mth*** *node.*
* *You also need to check conditions where ‘m’ is greater or smaller than ‘n’. (you can use object of linkedlist as parameter or without parameter it’s your own choice).*

**Type RotateAndDelete(Type &inputList, int n, int m)**

**OR**

**void RotateAndDelete(int n, int m)**

**Example 1:**

Input Linked List: **2 6 1 7 3 8 5 0**

INPUT: **m = 3, n = 7 (Now move 7th node at 3rd position, and then delete the 3rd node.**

OUTPUT:- **2 6 5 7 3 8 0**

**Example 2:**

Input Linked List: **2 6 1 7 3 8 5 0**

INPUT: **m = 7, n = 3 (Now move 3rd node at 7th position, and then delete the 7th node.**

OUTPUT:- **2 6 7 3 8 1 0**