# University of Central Punjab Faculty of Information Technology

**Data Structures and Algorithms Spring 2024**

|  |  |  |
| --- | --- | --- |
| **Lab 07** | |  |
| **Topic** | * Abstract Classes * Templates * Linear Queue, Circular Queue, Stack, Linked List * Applications of Queue, Stack, Linked List |
| **Objective** | The basic purpose of this lab is to implement Queues, Stack , Linked List and test its Applications |

**Instructions:**

* Indent 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 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.

**Task 1**

1. Create Struct **Node** which is having

***Attributes:***

* **Type data;**
* **Node\* next;**

1. Create abstract class named as **LinkedList**

***Attributes of Linked List:***

**Node \* head;**

**Node \*tail;**

Functions:

* **virtual void insertAtHead(Type data) = 0;**
* **virtual Type deleteFromHead() = 0;**
* **virtual void insertAtTail(Type data) = 0;**
* **virtual Type deleteFromTail() = 0;**
* **virtual bool isEmpty() = 0; //checks if linked list is empty or not**
* **virtual void display() = 0;**

Create **constructor** and **Destructor** for this class

Now you have to make 2 classes named as **Stack** and **Queue**

class **Stack**{

***Attributes:***

* **myLinkedList<Type> stk;**

***Functions:***

void push (Type data);

Type pop();

};

class **Queue**{

***Attributes:***

* **myLinkedList<Type> q;**

***Functions:***

void enqueuer(Type data);

Type dequeuer();

};

**Task 2**

Create an Abstract Class **Queue** with the following attributes and pure virtual functions

***Attributes:***

* **Type front;**
* **Type rear;**
* **int maxSize;**
* **Type\* arr;**

***Functions:***

* + **virtual bool isFull() = 0;**
  + **virtual bool isEmpty() = 0;**
  + **virtual void display() = 0;**
  + **virtual void enqueue(Type data) = 0;**
  + **virtual Type dequeue() = 0;**
  + **virtual Type size() = 0;**
  + **virtual Type peek() = 0;**

Create two functions in the main where we are required to perform the following:

***Functions in main():***

* **void displayOddNumbersInQueue(myQueue&);**
* **void displayEvenNumbersInQueue(myQueue&);**

After Implementation of the functions in myQueue create menu-based program to perform the following operations .:

1. **Check if queue is full or not**
2. **Check if queue is empty or not**
3. **Display queue**
4. **enqueue data in the Queue**
5. **dequeue data from Queue**
6. **Check size of the Queue**
7. **Display Peek value of the Queue**
8. **Display Odd Numbers in the Queue**
9. **Display Even Numbers in the Queue**
10. **Exit**