University of Central Punjab

**Faculty of Information Technology**

**Data Structures and Algorithms**

**Summers**

|  |  |  |
| --- | --- | --- |
| **Lab 06** | |  |
| **Topic** | * Abstract Classes * Templates * Linked List |
| **Objective** | * The basic purpose of this lab is to implement ADT of Linked List   and test its applications. |
|  | | |

**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.**

**Task 1**

Modify the code done in class and implement the linked list using **Head** pointer only (you are not allowed to use **Tail pointer).**

**Functions of Linked List:**

**front()** – Returns the value of the first element in the linked list.

**back()** – Returns the value of the last element in the linked list.

**insetAtFront(Type)** – Adds a new element at the beginning of the linked list.

**insetAtEnd(Type)** – Adds a new element at the end of the linked list**.**

**removeFromFront()** – Removes the first element of the linked list, and reduces size of the linked list by 1.

**RemoveFromEnd()** – Removes the last element of the linked list, and reduces size of the linked list by 1.

**empty()** – Returns whether the list is empty(1) or not(0).

**size()** – Returns the number of elements in the list.

**Task 2**

Implement **Stack** (LIFO) **and Queue** (FIFO) using **Linked Lists only**.

It should be a menu driven program which should first ask whether you want to use stack or queue. Then, operations related to stack or queue may be called.

**Task 3**

Implement an Insert function for linked list that inserts the data in sorted order.

