University of Central Punjab

**Faculty of Information Technology**

**Data Structures and Algorithms Spring 2023**

|  |  |  |
| --- | --- | --- |
| **Retake (1 hr)** | |  |
| **Topic** | * Circular 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**

Write a complete C++ code, where the head of a linked list is given, reverse the nodes of the list K at a time, and return the modified list. Kis a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of K then left-out nodes, in the end, should remain as it is. You may not alter the values in the list's nodes, only nodes themselves may be changed.

Input: 1->2->3->4->5; k=2

Output: 2->1->4->3->5

Input: 1->2->3->4->5; k=3

Output: 3->2->1->4->5

Input: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 and k = 3,  
output: 3 -> 2 -> 1 -> 6 -> 5 -> 4 -> 9 -> 8 -> 7, where the linked list is reversed in groups of 3 nodes.

Note:

* Use templates & make an Abstract Class.
* Write a function Node\* reverseNode(Node\* head, int k) for reversing the nodes. (You can’t swap the data).
* Make proper constructors & your code should be in running state. Straight ZERO will be awarded in case of errors.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***