Lahore University of Management and Sciences

Lab-10 Manual

for

Introduction to Programming

(CS200)

Dr. Mian Muhammad Awais

LAB GUIDELINES

- Make sure you submit the lab before 11:50 AM. Any late submission will not be graded afterwards. In case of internet connectivity or electricity issues make sure to email your assigned TA before 2:00 PM. **You should email your assigned TA ONLY**.
- For every lab, there will be a folder created on LMS. You must submit your work in the respective folder during the lab time, you and only you are responsible for your submissions.
- You will be allowed to discuss the questions in the first half of the lab session for a few questions. After that, there will be a portion of lab where you cannot converse and must work for yourself. No discussion is allowed in later time period.
- You should do your work with utmost clarity and precision. Do not waste your time trying to do something you do not understand. Ask Lab instructors for help, that is what they are there for.

- Any legitimate cheating case can and will be reported to Disciplinary Committee without any leniency. Plagiarism Software make our task easier.
- Please follow the lab etiquettes and follow code of conduct in the session.
- Do not start Personal chat during your zoom meeting and raise your hands before asking questions.
- **Make separate .cpp and .h files for each question. Naming convention for .cpp/.h files is: YourRollNumber_TaskX.cpp/.h. Before submission, copy all the .cpp/.h files in a folder named LabX_AssignedTAName_YourRollNumber.zip. Submit the .zip file only! (no .rar file submissions). X should be replaced by appropriate number.** Failure to follow the naming convention may lead to deduction of marks.

OBJECTIVES

- Singly Linked List

LAB EXERCISES

Question # 1 [Marks: 100] Est. Time: 60 mins

In this lab you will create a singly linked list of integer numbers. You are also required to make some relevant member functions for your singly linked list.

1- Create a struct for a node of the linked list **[5]**
2- Create a class for your linked list with appropriate private data members and the following member functions:

- constructor **[5]**
- A function that calculates the length of the list **[5]**
- A function that inserts a value to the head of the list **[10]**
- A function that inserts a value to the tail of the list **[10]**
- A function that inserts a value at any given position in the list. For example: consider a list 1->4->7, and you call the function **insertAtGivenPos(1, 3)** (where 1 is position and 3 is the value) then the function will alter the list to be like: 1->3->4->7 **[20]**
- A function that deletes the tail of the list **[5]**
- A function that deletes the head of the list **[5]**
- A function that deletes a node of the list at any given position **[20]**
- A function that prints the entire list **[5]**

You should implement a main function in which you will do the following (You **MUST** print your list after every step): **[10]**

- Insert the values 1, 2, 3, 4, 5 inside the list using any function (insert at head, insert at tail etc.) you want
- Insert 0 at head
- Insert 6 at the tail
- Insert 2 at the 2nd position (i.e. after 1 and before 2 in the current list state)
- Delete the tail
- Delete the head
- Delete the 3rd node (i.e. position 2) from the list

Note: You do not need to ask the user to input any values, all values should be hardcoded as specified.

Sample Output:

```
List after insertion: 1 -> 2 -> 3 -> 4 -> 5

List after insertion at head: 0 -> 1 -> 2 -> 3 -> 4 -> 5

List after insertion at tail: 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6

List after insertion at 2: 0 -> 1 -> 2 -> 2 -> 3 -> 4 -> 5 -> 6

List after delete tail: 0 -> 1 -> 2 -> 2 -> 3 -> 4 -> 5

List after delete head: 1 -> 2 -> 2 -> 3 -> 4 -> 5

List after deletion of node 3 (i.e. position 2): 1 -> 2 -> 3 -> 4 -> 5
```

Best of Luck!