|  |
| --- |
| **Data Structures and Algorithms** |
| **CL210** |
| **LABORATORY MANUAL** |
| **Fall 2021** |

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LAB 01** | | | | |
| **Design and Implementation of Link Lists and Its Operations using C++** | | | | |
| **Engr. Muhammad Adan** | | | | |
|  | |  | |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_ | | \_\_\_ |
| STUDENT NAME | | ROLL NO | | SEC |
|  | | | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |
| LAB ENGINEER SIGNATURE & DATE | | | | |
| **MARKS AWARDED:**  /**10** | | | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |
| **NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), ISLAMABAD** | | | | |
|  | | | | |
| Prepared by: | Engr. Shahid Qureshi |  | Date: 30 January 2018 | |
|  |  |  |  | |

|  |  |  |
| --- | --- | --- |
| **LAB:** | **02** | **Introduction to Linked Lists** |

#### **Learning Objectives:**

* To learn how to build a linked list, how to insert and delete new nodes in a linked list.

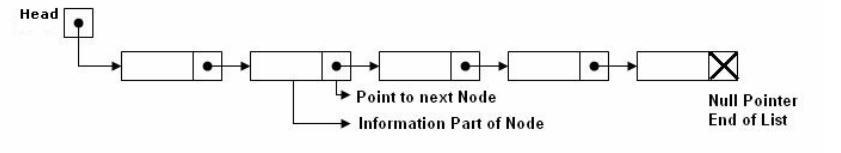
#### **Equipment Required:**

A working computer having Visual Studio or any other good compiler Installed.

1. **Introduction:**

Linked list is a linear collection of data elements, called nodes, where the linear order is given by means of “pointers”. Each node is divided into two parts.

* The first part contains the information of the element.
* The second part called the link field contains the address of the next node in the list.



**Advantages and Disadvantages**

* Linked lists are dynamic, so the length of a list can increase or decrease as necessary.
* Storage problems with fragmented memory
* Array requires contiguous space in memory. Array cannot dynamically grow incrementally. Must de-allocate and reallocate the whole array to change size.
* Linked list solves the above problems. Doesn't require contiguous memory. Linked list is a chain of pointer {connected nodes. Nodes store information just like array indices.
* Linked list access is slower than array. Must traverse the whole list to reach a particular node. But, storage and size is flexible.
* Array is preferable when speed is desired and storage size is known.

Example: Number of students in a class cannot exceed 50. Create an array of 50 size for students.

* Linked list is preferable when speed is not a major concern and storage size is highly dynamic. Example: Daily record of cars using motorway. On weekends traffic is high but there aren't many cars on weekdays. Numbers of cars can be highly varied.

**Basic Operations on Linked list**

* Adding a Node
* Deleting a Node
* Displaying Link List

**Lab Tasks**

1. Write a program to interact with a link list with the following menu
   * AppendAtEnd
   * DisplayAll
   * Destructor
   * Default Constructor
2. Make a function DeleteValue(float value), which should delete a node with the given value.
3. Make a function AddAtFront(), which should store the new value at first location. Use it to store some values, and see the result by calling the **DisplayAll** function.
4. Make a function insertAt(float value, int location), which should insert a new node at the given location.
5. Given a link list with some values (at least 10), write a program to find and print the sum, avg, min and max values.
6. Overload equality = = operator for the link list class, if the contents of the two link lists are same, then it should return true, else false
7. Copy Constructor(for when linked list is passed as an argument)