**National University of Computer and Emerging Sciences**



Laboratory Manual

for

Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Dr. Zareen Alamgir |
| Lab Instructor(s) | Fariha Maqbool  Humna Shabir |
| Section | BCS-3F |
| Semester | Fall 2022 |

**Department of Computer Science**

FAST-NU, Lahore, Pakistan

**Objectives:**

In this lab, students will practice:

1. Single Linked List

**Linked list Data Structure**

A linked list is a linear data structure that includes a series of connected nodes. Here, each node stores the data and the address of the next node. For example,

Linked list Data Structure

You have to start somewhere, so we give the address of the first node a special name called HEAD. Also, the last node in the linked list can be identified because its next portion points to NULL.

**Question 1**

1. Implement a template class ‘List’ with nested ‘Node’ struct that contains two data members: A template variable ‘data’ and a Node pointer ‘next’. You may define any member functions, if required, for the template class.
2. Now using the above class, implement a singly linked list which supports the following operations:
3. Insert at start void insertAtStart(T const element);
4. Insert at end void insertAtEnd(T const element);
5. Print void print() const;
6. Search an element bool search(T const& element) const;
7. Check whether the list is empty bool isEmpty() const;
8. Insert value v1 before value v2 bool insertBefore(T const v1, T const v2 ) const;
9. Print the linked list in reverse order void printrev() const;
10. Now create a main function which has the following instructions:
    1. Define a linked list object of type int.
    2. Insert 4, 7, and 9 at start
    3. Insert 2 at the end.
    4. Now insert 3, 7, and 1 at start.
    5. Now print the linked list.
    6. Search for 2, 9 and 7.
    7. Print the linked list in reverse