# 

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

**Task 1**

Create a C++ generic abstract class named as **LinkedList**, with the following:

**Attributes:**

1. Node \*head
2. Node \*tail

**Functions:**

virtual void insertAtTail(int) = 0;

* + Should add element at the tail node of linkedList.

virtual void deleteAtTail() = 0;

* + Should delete element at the tail node of linkedList.

LinkedList();

# Task 2

# LinkedList:

Linked List is type of data structure which contains some objects called nodes that are randomly stored in the memory. A node contains two fields i.e. data stored at that particular address and the pointer which contains the address of the next node in the memory. The last node of the list contains pointer to the null.

Use the class made in **Task 1** to make another class named as **MyLinkedList**, having following additional functionalities:

**void** [**display()**](https://www.geeksforgeeks.org/stack-empty-and-stack-size-in-c-stl/) : Should display the elements stored in linkedList.

1. Make a function which should calculate the sum of all values of linked list.

**void sum(Type &inputLinkedList)**

Example:

Linked List: 2 6 1 6 3

Sum of elements is: 18

1. Make a function to delete all the nodes having an odd value and display the final result.

**void removeOddValues(Type &inputLinkedList)**

Example:

Initial Linked List: 5 10 25 12 9 6 1 4 3

After removing odds: 10 12 6 4

# Implement both pure virtual functions insertAtTail() and deleteAtTail() declared in base in myLinkedList

After Implementation of the functions in myLinkedList, create menu based program to perform the following operations

1. Press 1 to add a new element to the tail of linkedlist. **void insertAtTail(int)**
2. Press 2 to delete the element from tail end of linkedlist. **void deleteAtTail()**
3. Press 3 to display the LinkedList.
4. Press 4 to calculate sum of all elements in LinkedList.
5. Press 5 to delete all the nodes having an odd value from LinkedList.
6. Press 0 to exit.
   * Write constructor for the above class.
   * Write Copy constructor for the above class.
   * Write Destructor for the above class.