

Amrita School of Engineering, Amritapuri Campus.

19CSE212: Data Structures and Algorithms

LAB SHEET 2-Singly Linked List

1. Create a class Singly Linked List with the following functions:
 - a. insertFirst() to insert an element at the beginning of the list
 - b. Insert the data at given position and call the function and debug your program.
 - c. insertLast() to insert an element at the end of the list.
 - d. isEmpty() and check if the list is empty or not.
 - e. Iteratively implement functions for deletions and check your program.
 - f. Sorted insertion
 - g. isLastElement()
2. Create driver class to perform the following :
 - a. Insert into the Linked List the values 10,20,30,40,60 in that order.
 - b. Delete nodes at position 3 and 5.
 - c. Insert into the existing List the value 25 using Sorted Insertion.
 - d. Delete 10 from the List
 - e. Check whether the list is empty.
 - f. Check whether the given element is the last element
 - g. Return the even positioned values

Note: Go through the skeleton code for your implementation

SKELETON CODE

```
class Node {
    int data;
    Node next;
    Node(int d) { data = d; next=null; } //Constructor to initialize data members
}

class LinkedList {
    Node head;
    public void printList() { //Function to print the elements
        Node n = head;
        System.out.print(n.data+" ");
        n = n.next;
    }

    public void insertAtFront(int data) {

        Node n = new Node(data);
        //dynamically allocate memory for the new node that is to be inserted
        .....
        .....

    }

    public void deleteFirst() {
        if (head==null)
        {
            System.out.println("List is Empty");
        }
        else
        {

        }

    }

}

public void insertAtLast(int data) {
    Node t=null; //t is a object of type node that is used for traversing
    Node n = new Node(data);
    if(head == null)
    {

    }

}
```

```

        else
        {
        }

    }

    public void deleteLast() {
        if(head == null)
        {
            System.out.println("List is empty");
            return;
        }
        else
        {
        }

    }

    public void insertAtPos(int data, int pos) {
        Node t=null;
        Node n = new Node(data);
        if(head == null && pos == 1)
        {

        }

    }

}

class Driver{

    public static void main(String[] args)           //main function
    {

        LinkedList l = new LinkedList();           //create object of the class LinkedList
        l.insertAtFront(1);
        l.insertAtFront(2);
        l.insertAtFront(3);
        .....
        .....

        l.printList();
    }           //close function main
}           //close class LinkedList

```