**Assignment 1:**

class Node

{

int data;

Node link;

Node(int d)

{

data = d;

link = null;

}

}

class LinkedList

{

Node head;

void Insert(Node new\_node)

{

Node current;

if (head == null || head.data >= new\_node.data)

{

new\_node.link = head;

head = new\_node;

}

else {

current = head;

while (current.link != null && current.link.data < new\_node.data)

current = current.link;

new\_node.link = current.link;

current.link = new\_node;

}

}

void delete(int data){

Node temp, pred ;

temp= head;

pred= head;

while(temp.link != null && temp.data != data)

{

pred=temp;

temp=temp.link;

}

if(temp.data== data)

pred.link=temp.link;

if(head.data== data)

head=temp.link;

}

Node newNode(int data)

{

Node x = new Node(data);

return x;

}

void display()

{

Node temp = head;

while (temp != null)

{

System.out.print(temp.data+" ");

temp = temp.link;

}

System.out.println();

}

public static void main(String args[])

{

LinkedList l = new LinkedList();

Node new\_node;

new\_node = l.newNode(2);

l.Insert(new\_node);

new\_node = l.newNode(7);

l.Insert(new\_node);

new\_node = l.newNode(1);

l.Insert(new\_node);

new\_node = l.newNode(5);

l.Insert(new\_node);

new\_node = l.newNode(6);

l.Insert(new\_node);

new\_node = l.newNode(4);

l.Insert(new\_node);

l.delete(4);

l.delete(1);

l.delete(7);

System.out.print("Linked List: ");

l.display();

}

}

**Assignment 2:**

public class Node {

public int data;

public Node blink,flink;

Node(int data){

this.data=data;

blink=null;

flink=null;

}

}

public class DoublyLinkedList {

Node blink, flink;

Node head;

void Insert(Node new\_Node){

Node newNode = new Node(new\_Node.data);

if(head==null)

{

head=newNode;

}

else if(new\_Node.data<head.data){

newNode.flink=head;

head=newNode;

}

else{

Node n1=null;

Node n2=head;

while (n2 != null && n2.data < new\_Node.data) {

n1 = n2;

n2 = n2.flink;

}

if (n2 == null)

{

n1.flink = newNode;

newNode.blink = n1;

}

else

{

n1.flink = newNode;

n2.blink = newNode;

newNode.blink = n1;

newNode.flink = n2;

}

}

}

void delete(int data){

Node temp, pred ;

temp= head;

pred= head;

while(temp.flink != null && temp.data != data)

{

pred=temp;

temp=temp.flink;

}

if(temp.data== data)

pred.flink=temp.flink;

if(head.data== data)

head=temp.flink;

}

void display()

{

Node temp = head;

while (temp != null)

{

System.out.print(temp.data+" ");

temp = temp.flink;

}

System.out.println();

}

Node newNode(int data)

{

Node x = new Node(data);

return x;

}

public static void main(String args[]){

DoublyLinkedList l = new DoublyLinkedList();

Node new\_node;

new\_node = l.newNode(2);

l.Insert(new\_node);

new\_node = l.newNode(7);

l.Insert(new\_node);

new\_node = l.newNode(1);

l.Insert(new\_node);

new\_node = l.newNode(5);

l.Insert(new\_node);

new\_node = l.newNode(6);

l.Insert(new\_node);

new\_node = l.newNode(4);

l.Insert(new\_node);

l.delete(7);

l.delete(1);

l.delete(5);

System.out.print("Linked List: ");

l.display();

}

}