

MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Topic: Unit 4: Lists 1) Singly Linked Lists 2) Circular Linked List (half)
Roll No. : MCA2511	Date : 03-11-2025

1) Demonstrate the working of a singly linked list with operations to insert, delete, search, display and count the number of nodes.

Code:

```
import java.util.Scanner;
```

```
//Class Node
```

```
class Node
```

```
{
```

```
    int data;
```

```
    Node next;
```

```
    public Node(int d)
```

```
    {
```

```
        data = d;
```

```
        next = null;
```

```
    } //end of Node DConstructor
```

```
} //end of class Node
```

```
//Class List
```

```
class List
```

```
{
```

```
    Node head;
```

```
    public List()
```

```
    {
```

```
        head = null;
```

```
    } //end of List DConstructor
```

```
    //Insert
```

```
    public void Insert(int x)
```

```
    {
```

```
        //Make a new Node
```

```
        Node t = new Node(x);
```

```
        //Fisrt node in SLL
```

```
        if(head==null)
```

```
{
    head = t;
    return;
}

//3. Traverse till last node and link t
Node tmp = head;
while(tmp.next!=null)
{
    tmp = tmp.next;
}
tmp.next = t;
} //end of Insert

//Delete
public void Delete(int x)
{
    //1. Search for x
    boolean flag = false;
    Node tmp = head;
    Node prev = null;

    while(tmp!=null)
    {
        if(tmp.data == x)
        {
            flag = true;
            break;
        }
        prev = tmp;
        tmp = tmp.next;
    }

    //2. x not found
    if(flag==false)
    {
        System.out.println(x + " is not found");
        return;
    }

    //3. x is found
    if(tmp==head && tmp.next==null)    //3a. Single Node deletion
    {
        head = null;
    }
    else if(tmp == head)                //3b. Head Node Deletion
```

```
{
    head = tmp.next;
}
else if(tmp.next == null)           //3c. Tail Node Deletion
{
    prev.next = null;
}
else                               //3d. Any other node in the middle
{
    prev.next = tmp.next;
}
System.out.println("Deleted Element Successfully");

} //end of Delete

//Search
public boolean Search(int x)
{
    Node tmp = head;
    boolean flag = false;
    while(tmp != null)
    {
        if(tmp.data == x)
        {
            flag = true;
            break;
        }
        tmp = tmp.next;
    }
    return flag;
} //end of Search

//Count
public void Count()
{
    Node tmp = head;
    int count = 0;
    while(tmp != null)
    {
        count++;
        tmp = tmp.next;
    }
    System.out.println("No. of Nodes are " + count );
} //end of Count
```

```
//Display
public void Display()
{
    System.out.println("SLL contains: ");
    Node tmp = head;
    while(tmp !=null)
    {
        System.out.print(tmp.data + "->");
        tmp = tmp.next;
    }
}

} //end of Display

} //end of class List

//Main Class SLL
class SLL
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        List s = new List();

        int ch;

        do
        {
            System.out.println("\n-----");
            System.out.println(" Singly Linked List Menu");
            System.out.println("-----");
            System.out.println("1. Insert at the end of SLL");
            System.out.println("2. Delete an element");
            System.out.println("3. Search for an element");
            System.out.println("4. Count the number of nodes in SLL");
            System.out.println("5. Display the SLL");
            System.out.println("6. Exit\n");

            System.out.print("Enter your Choice: ");
            ch = sc.nextInt();

            switch(ch)
            {
                case 1:
                    System.out.print("Enter a Value: ");
                    int x = sc.nextInt();
                    s.Insert(x);
                    break;
```

```
case 2:
    System.out.print("Enter a Value: ");
    x = sc.nextInt();
    s.Delete(x);
break;

case 3:
    System.out.print("Enter a Value: ");
    x = sc.nextInt();
    if(s.Search(x))
    {
        System.out.println("Data is found");
    }
    else
    {
        System.out.println("Data is not found");
    }
break;

case 4:
    s.Count();
break;

case 5:
    s.Display();
break;

case 6:
    System.out.println("Exiting Program..... :)");
break;

default:
    System.out.println("Incorrect Choice Try again");
break;
    }

    }while(ch!=6);
} //end of psvm
} //end of class SLL
```

Output:

```
A:\MCA2511\DS_LAB>javac 10SLL.java
```

```
A:\MCA2511\DS_LAB>java SLL
```

```
-----  
Singly Linked List Menu  
-----
```

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

```
Enter your Choice: 1
```

```
Enter a Value: 50
```

```
-----  
Singly Linked List Menu  
-----
```

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

```
Enter your Choice: 1
```

```
Enter a Value: 60
```

```
-----  
Singly Linked List Menu  
-----
```

1. Insert at the end of SLL

3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 1
Enter a Value: 80

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 1
Enter a Value: 60

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 1
Enter a Value: 30

3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 1

Enter a Value: 30

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 5

SLL contains:

50->60->80->60->30

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 6

Exiting Program..... :)

A:\MCA2511\DS_LAB>

Enter your Choice: 5
SLL contains:
10->20->30->40->50->

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 3
Enter a Value: 20
Data is found

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 3
Enter a Value: 25
Data is not found

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 4
No. of Nodes are 5

Enter your Choice: 5
SLL contains:
10->20->30->40->50->

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 2
Enter a Value: 30
Deleted Element Successfully

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 5
SLL contains:
10->20->40->50->

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 2
Enter a Value: 13
13 is not found

Enter your Choice: 2
Enter a Value: 10
Deleted Element Successfully

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 2
Enter a Value: 20
Deleted Element Successfully

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 2
Enter a Value: 40
Deleted Element Successfully

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 2
Enter a Value: 50
Deleted Element Successfully

Singly Linked List Menu

1. Insert at the end of SLL
2. Delete an element
3. Search for an element
4. Count the number of nodes in SLL
5. Display the SLL
6. Exit

Enter your Choice: 5
SLL contains:

2) Demonstrate the working of a Circular linked list with an operations menu to insert, delete, search, display and count the number of nodes.

Code:

```
import java.util.Scanner;

//CNode Template
class CNode
{
    int data;
    CNode next;

    public CNode(int d)
    {
        data = d;
        next = null;
    }
}

//end of CNode

//CList Template
class CList
{
    CNode head;
    CNode tail;

    public CList()
    {
        head = null;
        tail = null;
    }

    //Insert

    //Delete

    //Search

    //Count

    //Delete
}

//end of CList

//CLL
class CLL
{
    public static void main(String[] args)
```

```
{
    Scanner sc = new Scanner(System.in);
    CList c = new CList();
    int ch;

    do
    {
        System.out.println("\n---Circular Linked List---");
        System.out.println("1. Insert a node in CLL");
        System.out.println("2. Delete a node in CLL");
        System.out.println("3. Search for a Node in CLL");
        System.out.println("4. Display the CLL");
        System.out.println("5. Count No. of Nodes in CLL");
        System.out.println("6. Exit");

        System.out.print("Enter your choice: ");
        ch = sc.nextInt();

        switch(ch)
        {

            case 1:
                break;

            case 2:
                break;

            case 3:
                break;

            case 4:
                break;

            case 5:
                break;

            case 6:
                System.out.println("Exiting..... :");
                break;

            default:
                System.out.println("Incorrect Choice... Try again...");

        } //end of switch

    } while (ch != 6);    //end of do-while
```

```
    }//end of psvm  
}//end of class
```

Output:

```
A:\MCA2511\DS_LAB>java CLL
```

```
----Circular Linked List----
```

1. Insert a node in CLL
2. Delete a node in CLL
3. Search for a Node in CLL
4. Display the CLL
5. Count No. of Nodes in CLL
6. Exit

```
Enter your choice: 12
```

```
Incorrect Choice... Try again...
```

```
----Circular Linked List----
```

1. Insert a node in CLL
2. Delete a node in CLL
3. Search for a Node in CLL
4. Display the CLL
5. Count No. of Nodes in CLL
6. Exit

```
Enter your choice: 6
```

```
Exiting..... :)
```

```
A:\MCA2511\DS_LAB>
```