

MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Topic: Unit 4: Lists (i) Circular Linked List Deletion (ii) Doubly Linked List
Roll No. : MCA2511	Date : 10-11-2025

- 1) Demonstrate the working of a Circular linked list with operations 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
    public void Insert(int x)
    {
        //1. Make a new node
        CNode t = new CNode(x);
```

```

//2. First node in the CLL
if(head == null)
{
    head = t;
    tail = t;
    tail.next = head;
}
else //3. Any other node - insert at the end of CLL
{
    tail.next = t;           //Connect tail to t
    tail = t;               //Update tail
    tail.next = head;       //Update Circularity
}
}

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

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

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

    //3. x is found
    if(tmp == head && tmp == tail)      //3a. Single Node Deletion
    {
        head = null;
    }
}

```

```
tail = null;
}
else if (tmp==head)           //3b. Head Node Deletion
{
    head = tmp.next;
    tail.next = head;
}
else if (tmp==tail)           //3c. Tail Node Deletion
{
    tail = prev;
    tail.next = head;
}
else                         //3d. Any other node
{
    prev.next = tmp.next;
}
}

//Search
public void Search(int x)
{
    CNode tmp = head;
    boolean flag = false;
    if(tmp == null)
    {
        System.out.println("Empty CLL");
        return;
    }
    do
    {
        if(tmp.data == x)
        {
            flag = true;
        }
        tmp = tmp.next;
    } while (tmp!=head);

    if(flag)
    {
        System.out.println("Element Found :)");
    }
    else
    {
        System.out.println("Element not Fount :(");
    }
}
```

```
//Count
public void Count()
{
    CNode tmp = head;
    int count = 0;
    if(tmp == null)
    {
        System.out.println("Empty CLL");
        return;
    }
    do
    {
        count++;
        tmp = tmp.next;
    } while (tmp!=head);
    System.out.print("No. of Nodes are "+count);
}

//Display
public void Display()
{
    CNode tmp = head;
    if(tmp == null)
    {
        System.out.println("Empty CLL");
        return;
    }
    System.out.print("Circular Linked List contains ");
    do
    {
        System.out.print(tmp.data + " -> ");
        tmp = tmp.next;
    } while (tmp!=head);
    System.out.print("Back to head\n");
}

}//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. Count No. of Nodes in CLL");
    System.out.println("5. Display the CLL");
    System.out.println("6. Exit");

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

    switch(ch)
    {

        case 1:
            System.out.print("Enter a Value: ");
            int x = sc.nextInt();
            c.Insert(x);
            c.Display();
            break;

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

        case 3:
            System.out.print("Enter a Element to find: ");
            x = sc.nextInt();
            c.Search(x);
            break;

        case 4:
            c.Count();
            break;

        case 5:
            c.Display();
            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:

Insertion

```
A:\MCA2511\DS_LAB>javac 12CLL.java
```

```
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. Count No. of Nodes in CLL
5. Display the CLL
6. Exit

```
Enter your choice: 1
```

```
Enter a Value: 22
```

----Circular Linked List----

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

```
Enter your choice: 1
```

```
Enter a Value: 33
```

----Circular Linked List----

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

```
Enter your choice: 1
```

```
Enter a Value: 44
```

----Circular Linked List----

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

Enter your choice: 1

Enter a Value: 55

Display:

----Circular Linked List----

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

Enter your choice: 5

Circular Linked List contains 22 -> 33 -> 44 -> 55 -> Back to head

Count:

----Circular Linked List----

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

Enter your choice: 4

No. of Nodes are 4

Search:

Circular Linked List contains 22 -> 33 -> 44 -> 55 -> Back to head

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

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

Enter your choice: 3

Enter a Element to find: 22

Element Found :)

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

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

Enter your choice: 3

Enter a Element to find: 121

Element not Fount :(

Add more 66,77,88

----Circular Linked List----

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

Enter your choice: 1

Enter a Value: 66

Circular Linked List contains 22 -> 33 -> 44 -> 55 -> 66 -> Back to head

----Circular Linked List----

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

Enter your choice: 1

Enter a Value: 77

Circular Linked List contains 22 -> 33 -> 44 -> 55 -> 66 -> 77 -> Back to head

----Circular Linked List----

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

Enter your choice: 1

Enter a Value: 88

Circular Linked List contains 22 -> 33 -> 44 -> 55 -> 66 -> 77 -> 88 -> Back to head

Deletion of Head Node:

```
----Circular Linked List----  
1. Insert a node in CLL  
2. Delete a node in CLL  
3. Search for a Node in CLL  
4. Count No. of Nodes in CLL  
5. Display the CLL  
6. Exit  
Enter your choice: 2  
Circular Linked List contains 22 -> 33 -> 44 -> 55 -> 66 -> 77 -> 88 -> Back to head  
Enter a Value: 22  
Circular Linked List contains 33 -> 44 -> 55 -> 66 -> 77 -> 88 -> Back to head
```

Deletion of Tail Node:

```
----Circular Linked List----  
1. Insert a node in CLL  
2. Delete a node in CLL  
3. Search for a Node in CLL  
4. Count No. of Nodes in CLL  
5. Display the CLL  
6. Exit  
Enter your choice: 2  
Circular Linked List contains 33 -> 44 -> 55 -> 66 -> 77 -> 88 -> Back to head  
Enter a Value: 88  
Circular Linked List contains 33 -> 44 -> 55 -> 66 -> 77 -> Back to head
```

Deletion of Any other Node:

```
----Circular Linked List----  
1. Insert a node in CLL  
2. Delete a node in CLL  
3. Search for a Node in CLL  
4. Count No. of Nodes in CLL  
5. Display the CLL  
6. Exit  
Enter your choice: 2  
Circular Linked List contains 33 -> 44 -> 55 -> 66 -> 77 -> Back to head  
Enter a Value: 55  
Circular Linked List contains 33 -> 44 -> 66 -> 77 -> Back to head
```

- 2) Demonstrate the working of a Doubly linked list with operations to insert, search, display and count the number of nodes.

Code:

```
import java.util.Scanner;

class DNode
{
    int data;
    DNode prev;
    DNode next;

    DNode(int d)
    {
        data = d;
        prev = null;
        next = null;
    }
}//end of DNode

//DList
class DList
{
    DNode head;
    DNode tail;

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

    //Insert
    public void Insert(int x)
    {
        //1. Make a new node t
        DNode t =new DNode(x);

        //2. First node in DLL
        if(head == null)
        {
            head = t;
            tail = t;
        }
        else //3. Link t at the end of the DLL
        {
    }
```

```

tail.next = t;
t.prev = tail;
tail = t;
}
}//end of Insert

//Delete

//Search
public void Search(int x)
{
    DNode tmp = head;
    boolean flag = false;

    while(tmp!=null)
    {
        if(tmp.data == x)
        {
            flag = true;
            break;
        }
        tmp = tmp.next;
    }
    if(flag==true)
    {
        System.out.println(x+ " is found");
    }
    else
    {
        System.out.println(x+ " is not found");
    }
}//end of Search

//Count
public void Count()
{
    DNode tmp = head;
    int cnt = 0;

    while(tmp!=null)
    {
        cnt++;
        tmp = tmp.next;
    }
    System.out.println("There are "+ cnt + " nodes");
}//end of Count

```

```

//Display
public void Display()
{
    DNode tmp = head;
    System.out.print("From head: ");
    while(tmp!=null)
    {
        System.out.print(tmp.data + "<->");
        tmp = tmp.next;
    }
    System.out.print("NULL");
    System.out.println();

    tmp = tail;
    System.out.print("From tail: ");
    while(tmp!=null)
    {
        System.out.print(tmp.data + "<->");
        tmp = tmp.prev;
    }
    System.out.print("NULL\n");
}//end of Display

}//end of DList

class DLL
{
    public static void main(String[] args)
    {
        int ch;
        DList d = new DList();
        Scanner sc = new Scanner(System.in);

        do
        {
            System.out.println("\n*****Doubly Linked List*****");
            System.out.println("1. Insert a new node in DLL");
            System.out.println("2. Delete a node in DLL");
            System.out.println("3. Search for a node in DLL");
            System.out.println("4. Count No. of Nodes in DLL");
            System.out.println("5. Display Nodes in DLL");
            System.out.println("6. Exit out");
            System.out.println();
    
```

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

switch(ch)
{
    case 1:
        System.out.print("Enter a value: ");
        int x = sc.nextInt();
        d.Insert(x);
        d.Display();
        break;

    case 2:
        break;

    case 3:
        System.out.print("Enter a value: ");
        x = sc.nextInt();
        d.Search(x);
        break;

    case 4:
        d.Count();
        break;

    case 5:
        d.Display();
        break;

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

    default:
        System.out.println("Incorrect choice :c \n");
        break;
}

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

Output:

```
A:\MCA2511\DS_LAB>javac 14DLL.java
```

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

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

```
Enter your choice: 1
```

```
Enter a value: 22
```

```
From head: 22<->NULL
```

```
From tail: 22<->NULL
```

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

```
Enter your choice: 1
```

```
Enter a value: 33
```

```
From head: 22<->33<->NULL
```

```
From tail: 33<->22<->NULL
```

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 1

Enter a value: 44

From head: 22<->33<->44<->NULL

From tail: 44<->33<->22<->NULL

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 1

Enter a value: 55

From head: 22<->33<->44<->55<->NULL

From tail: 55<->44<->33<->22<->NULL

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 1

Enter a value: 66

From head: 22<->33<->44<->55<->66<->NULL

From tail: 66<->55<->44<->33<->22<->NULL

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 4

There are 5 nodes

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 5

From head: 22<->33<->44<->55<->66<->NULL

From tail: 66<->55<->44<->33<->22<->NULL

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 3

Enter a value: 12

12 is not found

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 3

Enter a value: 55

55 is found

*****Doubly Linked List*****

1. Insert a new node in DLL
2. Delete a node in DLL
3. Search for a node in DLL
4. Count No. of Nodes in DLL
5. Display Nodes in DLL
6. Exit out

Enter your choice: 6

Exiting ----- :