MCA Semester 1	Subject : Advanced Data Structures Lab
Name : Mukund Gangurde	Topic: Unit 4: Lists Circular Linked List
Roll No. : MCA2511	Date : 07-11-2025

1) Demonstrate the working of a Circular linked list with operations to insert, 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;
	}//end of CList

	//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
        }
}//end of Insert

//Delete

//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 :(");
        }
}//end of Search

//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);
	}//end of 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");
	}//end of Display
}//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);
                break;

                case 2:

                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 :(
```