| Program Name: | 08\_1 |
| --- | --- |
| Roll No: | 1510 |
| Title of Program: | Circular Linked List |
| Objective: | Code for Circular Linked list with implementation of:   1. Insert 2. Search 3. Count 4. Display 5. Deletion |

**CODE:**

/\*

Name: Advait Dhakad

Roll no: 1510

Unit 4: Lists

Program: Circular Linked List \*/

import java.util.Scanner;

/\* 1. Node Template \*/

class CNode{

int data;

CNode next;

/\* Constructor \*/

public CNode(int d)

{

data = d;

next = null;

}//End of Constructor

} //end of Node

class Clist{

CNode head;

CNode tail;

Clist(){

head=null;

tail=null;

}

// insert at the end of the list

void insert(int val){

CNode x = new CNode(val);

if(head == null){

head = x;

} // end of if

else{

tail.next=x;

} // end of else

tail=x;

tail.next=head;

System.out.println("Node added!!");

} // end of Cinsert

// display the cll

void Display(){

if(head==null){

System.out.println("EMPTY LIST!!!");

}//end of if

else{

CNode tmp = head;

do{

System.out.print(" " + tmp.data+ "->");

tmp = tmp.next;

}while(tmp!=head); // end of do-while

System.out.print("End of list \n");

} // end of else

} // end of Display

// count the number of nodes

int count\_nodes(){

if(head==null){

return 0;

}//end of if

else{

int count = 0;

CNode tmp = head;

do{

count ++;

tmp = tmp.next;

}while(tmp!=head); // end of do-while

return count;

} // end of else

} // end of Count nodes

// search for the node

boolean search(int val){

if(head==null){

System.out.println("EMPTY LIST!!!");

return false;

}//end of if

else{

int flg = 0;

CNode tmp = head;

do{

if(tmp.data==val){

flg=1;

break;

} // end of searching if

tmp = tmp.next;

}while(tmp!=head); // end of do-while

return flg == 1 ? true : false;

} // end of else

} // end of Search

// delete ccode

void delete(int ele){

boolean flg = search(ele);

if (flg==false){

System.out.println("The Data not found:");

return;

}

CNode tmp = head;

CNode prev = null;

do{

if(tmp.data==ele){

break;

} // end of searching if

tmp = tmp.next;

prev = tmp;

}while(tmp!=head); // end of do-while

if(tmp == head && tmp==tail){

head = null;

tail = null;

}

else if(tmp == head){

head= head.next;

tail.next= head;

}

else if(tmp == tail){

tail = prev;

tail.next=head;

}

else{

prev.next=tmp.next;

}

} // end of delete

} //end of Clist

class CLL{

public static void main(String[] args){

Scanner scan = new Scanner(System.in);

char ch;

Clist c = new Clist();

do{

System.out.print("\033[H\033[2J");

System.out.flush();

System.out.println("\t\t\*\*\*\*\*\*\*\*CIRCULAR LINKED LIST\*\*\*\*\*\*\*\*\*");

System.out.println(" 1 . Insert at the end of the CLL");

System.out.println(" 2 . Count the number of nodes");

System.out.println(" 3 . Display the CLL");

System.out.println(" 4 . Search for a Node");

System.out.println(" 5 . Delete a node \n");

System.out.print("Enter your Choice: ");

int choice = scan.nextInt();

switch(choice){

case 1:

System.out.println("Selected to insert the data");

System.out.print("Enter the data you want insert: ");

c.insert(scan.nextInt());

break;

case 2:

System.out.println("Selected to Count the number of nodes");

System.out.println("The Circular Link List has "+c.count\_nodes()+" elements.");

break;

case 3:

System.out.println("Selected to Display the CLL");

c.Display();

break;

case 4:

System.out.println("Selected to Search for a node");

System.out.print("Enter the data you want Search:");

System.out.println(c.search(scan.nextInt())==true? "The element is present.": "The element is not present.");

break;

case 5:

System.out.println("Selected to Delete a node");

c.Display();

System.out.print("Enter the element you want to delete: ");

c.delete(scan.nextInt());

System.out.println("\nAfter Deletion: ");

c.Display();

break;

default:

System.out.println("Wrong choice!!");

break;

}// end of switch

System.out.print("Do you Want to Countinue(y or Y for yes):" );

ch = scan.next().charAt(0);

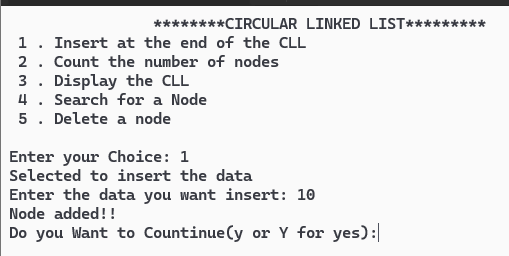
}while(ch=='y' || ch=='Y'); // end of do while

} //end of main

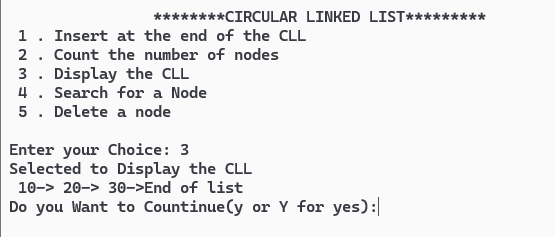
} // end of CLL

**OUTPUT:**

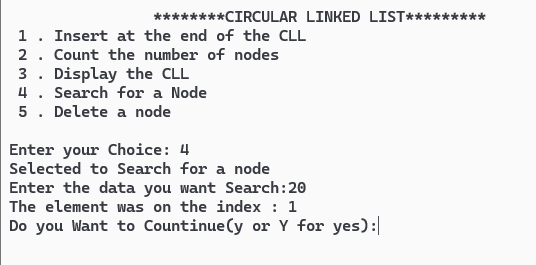
ADDING Node:



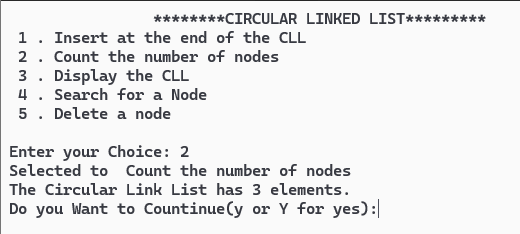
DISPLAY:

****

SEARCH:



COUNT:



DELETE:

