1. Write a Java program to implement Circular Linked List Using Array And Class

**package** prog14;

**public class** CircularLinkedList {

**public int** size =0; **public** Node head=**null**; **public** Node tail=**null**;

//add a new node at the start of the linked list

**public void** addNodeAtStart(**int** data){ System.***out***.println("Adding node " + data + " at start"); Node n = **new** Node(data);

**if**(size==0){

**head = n;**

**tail = n;**

**n.next = head;**

**}else{**

Node temp = head; n.next = temp;

head = n;

tail.next = head;

}

size++;

}

**public void** addNodeAtEnd(**int** data){

**if**(size==0){

addNodeAtStart(data);

}**else**{

Node n = **new** Node(data);

tail.next =n;

tail=n;

tail.next =n;

head;

size++;

}

System.***out***.println("\nNode " + data + " is added at the end of the list");

}

**public void** deleteNodeFromStart(){

**if**(size==0){

System.***out***.println("\nList is Empty");

}**else**{

System.***out***.println("\ndeleting node " + head.data + " from start");

head = head.next;

tail.next=head; size--;

}

}

**public int** elementAt(**int** index){

**if**(index>size){

**return** -1;

}

Node n = head;

**while**(index-1!=0){ n=n.next; index--;

}

**return** n.data;

}

//print the linked list

**public void** print(){ System.***out***.print("Circular Linked List:"); Node temp = head;

**if**(size<=0){

System.***out***.print("List is empty");

}**else**{

**do** {

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

temp = temp.next;

}

**while**(temp!=head);

}

System.***out***.println();

}

//get Size

**public int** getSize(){

**return** size;

**public static void** main(String[] args) {

CircularLinkedList c = **new** CircularLinkedList();

c.addNodeAtStart(3);

c.addNodeAtStart(2); c.addNodeAtStart(1); c.print(); c.deleteNodeFromStart(); c.print(); c.addNodeAtEnd(4); c.print();

System.***out***.println("Size of linked list: "+ c.getSize()); System.***out***.println("Element at 2nd position: "+ c.elementAt(2));

}

}

**package** prog14;

**class** Node{

**int** data; Node next;

**public** Node(**int** data){

**this**.data = data;

}

}

Output:

