DAILY ONLINE ACTIVITIES SUMMARY

Date:	**03-06-2020**	**Name:**	**Anvitha Poojary**
Sem & Sec	**6A**	**USN:**	**4AL17CS008**

Online Test Summary

Subject	**PAP assignment test**		
Max. Marks	**20**	**Score**	**9**

Certification Course Summary

Course	**Python for data science**		
Certificate Provider	**COGNITIVE CLASS .ai**	**Duration**	**5hr**

Coding Challenges

Problem Statement:

1.Take a list of length 3 containing integers, find out which is larger, first or last one and set all the elements in the list to be that value. Print the updated list

2.Write a python program to generate prime number in an interval

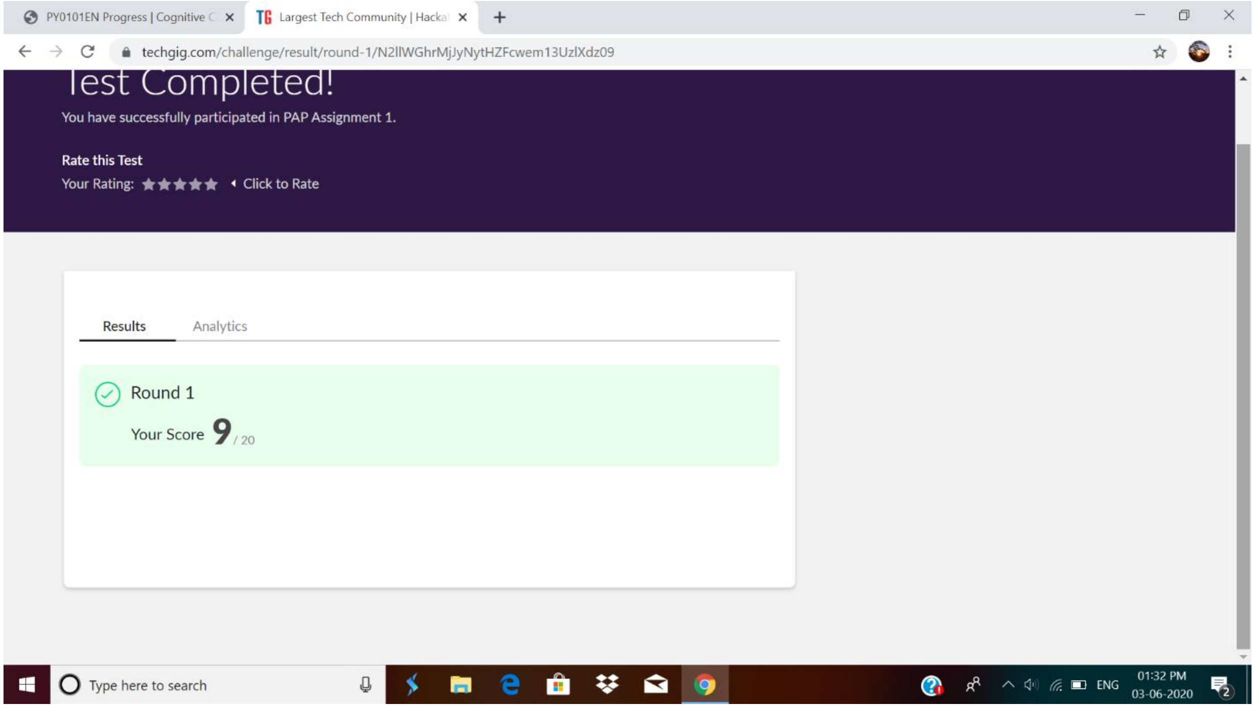
3. Write a Java Program to Implement Circular Doubly Linked List

Status: completed

Uploaded the report in Github	**Yes**
If yes Repository name	https://github.com/anvithapo99/Daily-Report
Uploaded the report in slack	**Yes**

Online test details:

Subject: PAP



PY0101EN Progress | Cognitive C
Largest Tech Community | Hacka
techgig.com/challenge/result/round-1/N2llWGhrMjJyNytHZFcwem13UzlXdz09

Test Completed!

You have successfully participated in PAP Assignment 1.

Rate this Test

Your Rating: ★★★★★ Click to Rate

Round 1

Your Score 9 / 20

Type here to search
ENG
01:32 PM
03-06-2020

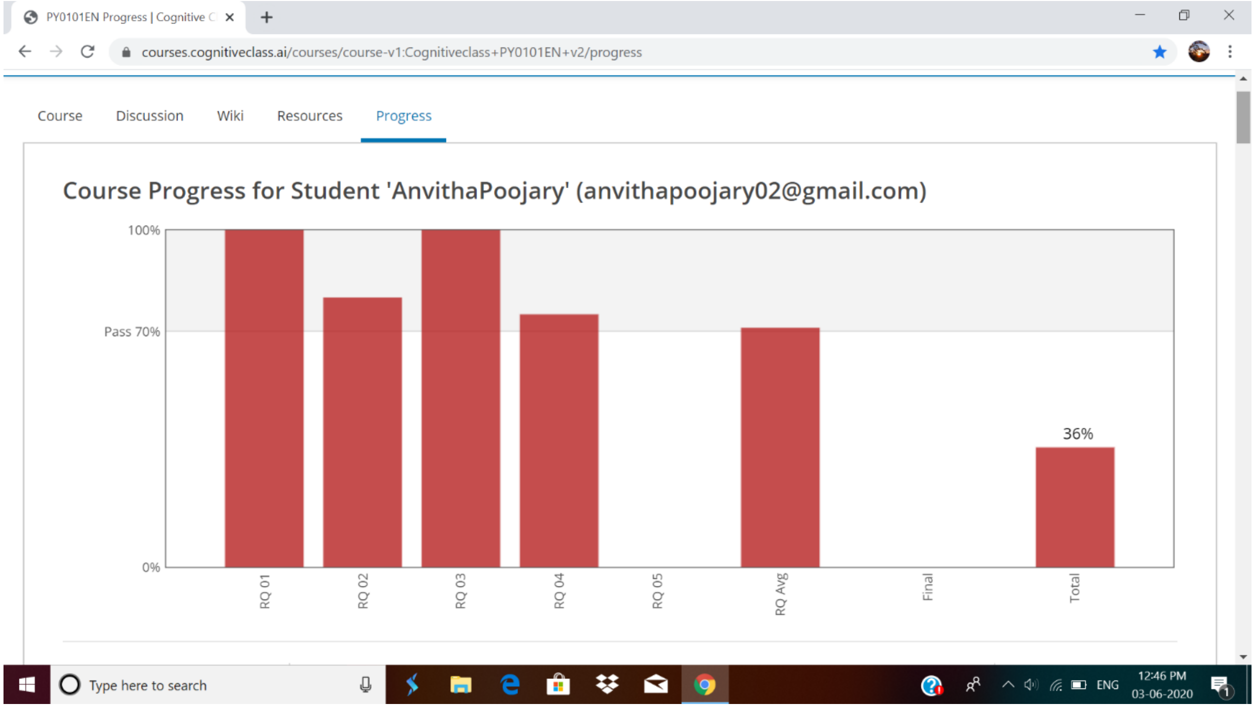
Certification course details:

Python for data science

Today I have studied following topics:

- Reading data from file
- Writing data to file
- Appending data to file
- Some example program
- Data frames
- Some methods of dataframe

- And its example program



PY0101EN Progress | Cognitive C
courses.cognitiveclass.ai/courses/course-v1:Cognitiveclass+PY0101EN+v2/progress
Course
Discussion
Wiki
Resources
Progress

Course Progress for Student 'AnvithaPoojary' (anvithapoojary02@gmail.com)

Coding Challenges Details:

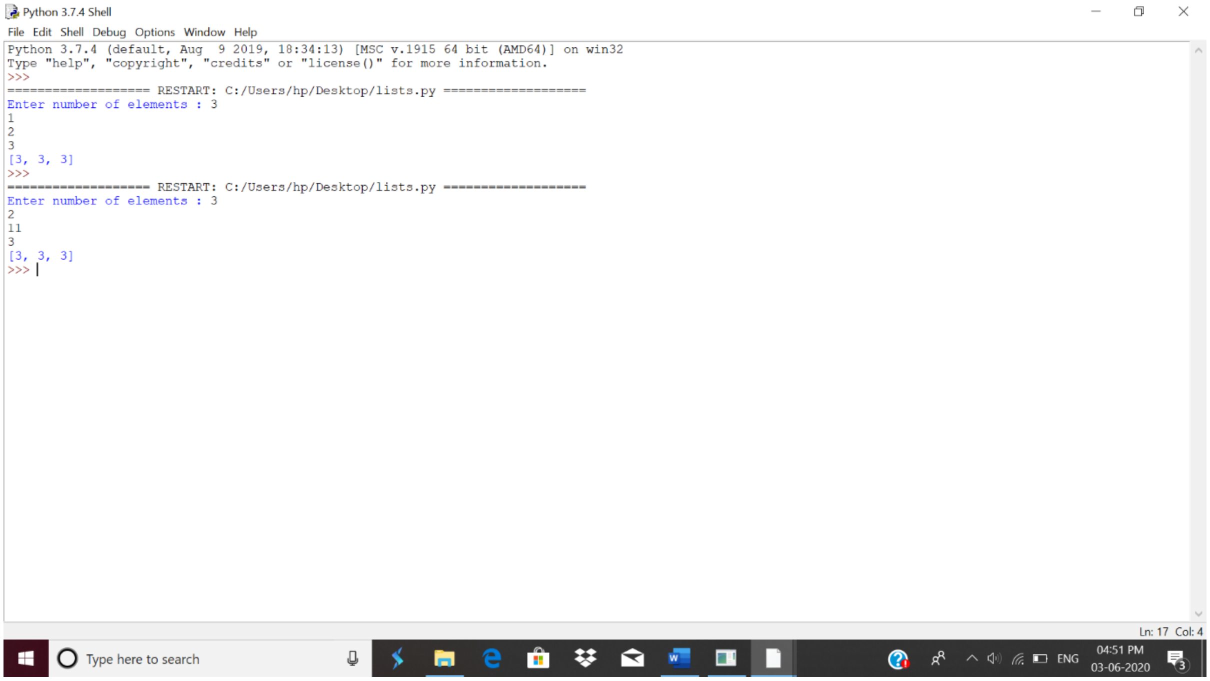
1. Python Program

Problem statement:
Take a list of length 3 containing integers, find out which is larger, first or last one and set all the elements in the list to be that value. Print the updated list
eg:
1)Input - Given list: [1, 2, 3]
Output- [3,3,3]
2)Input - Given list: [2, 11, 3]
Output- [3,3,3]

```
lst=[]
```

```
n = int(input("Enter number of elements : "))

for i in range(0, n):

    ele = int(input())



    lst.append(3)



print(lst)
```

output:

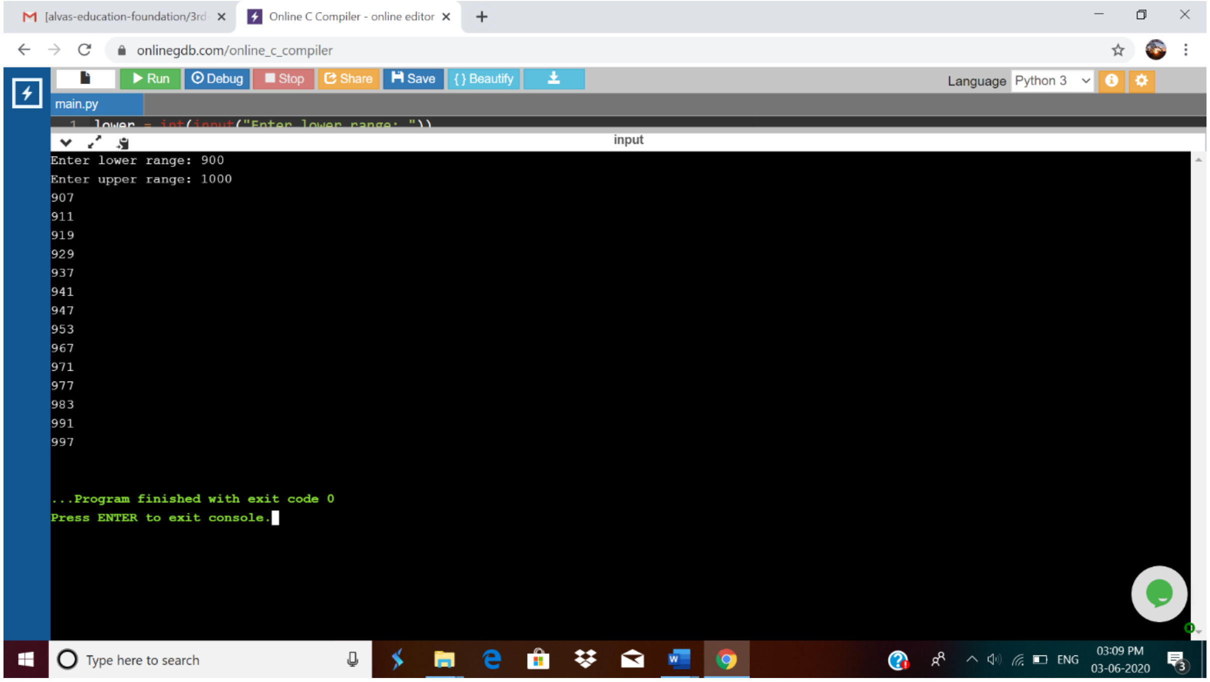


```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: C:/Users/hp/Desktop/lists.py ===================
Enter number of elements : 3
1
2
3
[3, 3, 3]
>>>
=================== RESTART: C:/Users/hp/Desktop/lists.py ===================
Enter number of elements : 3
2
11
3
[3, 3, 3]
>>> 
```

2.Write a python program to generate prime number in an interval

```
lower = int(input("Enter lower range: "))

upper = int(input("Enter upper range: "))



for num in range(lower,upper + 1):
 if num > 1:
    for i in range(2,num):
        if (num % i) == 0:
            break
    else:
        print(num)
```

output:



```
Enter lower range: 900
Enter upper range: 1000
907
911
919
929
937
941
947
953
967
971
977
983
991
997


...Program finished with exit code 0
Press ENTER to exit console.
```

3. Write a Java Program to Implement Circular Doubly Linked List

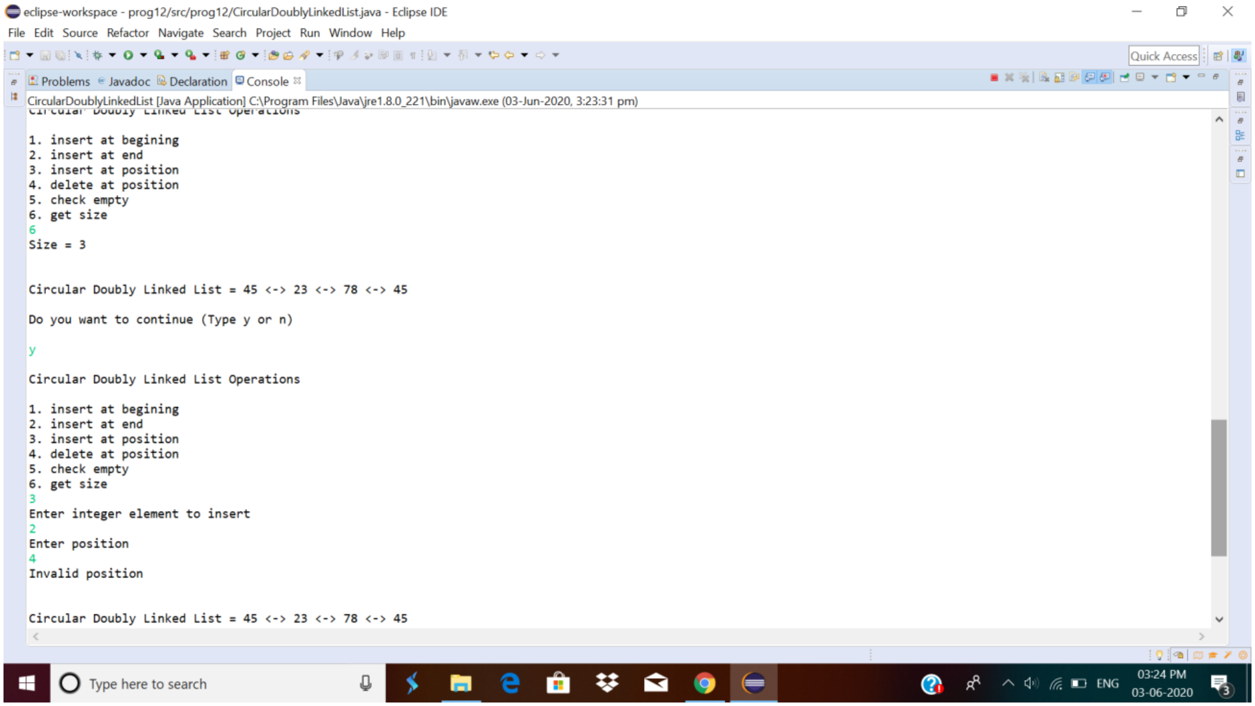
```
package prog12;

import java.util.Scanner;
class Node
{
    protected int data;
    protected Node next, prev;

    public Node()
    {
        next = null;
        prev = null;
```

```java
            data = 0;
        }
        public Node(int d, Node n, Node p)
        {
            data = d;
            next = n;
            prev = p;
        }

        public void setLinkNext(Node n)
        {
            next = n;
        }

        public void setLinkPrev(Node p)
        {
            prev = p;
        }

        public Node getLinkNext()
        {
            return next;
        }
        /* Function to get link to previous node */
        public Node getLinkPrev()
        {
            return prev;
        }
        /* Function to set data to node */
        public void setData(int d)
        {
            data = d;
        }
        /* Function to get data from node */
        public int getData()
        {
            return data;
        }
    }

    /* Class linkedList */
    class linkedList
    {
        protected Node start;
        protected Node end ;
        public int size;
        public linkedList()
        {
            start = null;
            end = null;
            size = 0;
        }
        public boolean isEmpty()
        {
            return start == null;
```

```
}

public int getSize()
{
    return size;
}
public void insertAtStart(int val)
{
    Node nptr = new Node(val, null, null);
    if (start == null)
    {
        nptr.setLinkNext(nptr);
        nptr.setLinkPrev(nptr);
        start = nptr;
        end = start;
    }
    else
    {
        nptr.setLinkPrev(end);
        end.setLinkNext(nptr);
        start.setLinkPrev(nptr);
        nptr.setLinkNext(start);
        start = nptr;
    }
    size++ ;
}
/*Function to insert element at end */
public void insertAtEnd(int val)
{
    Node nptr = new Node(val, null, null);
    if (start == null)
    {
        nptr.setLinkNext(nptr);
        nptr.setLinkPrev(nptr);
        start = nptr;
        end = start;
    }
    else
    {
        nptr.setLinkPrev(end);
        end.setLinkNext(nptr);
        start.setLinkPrev(nptr);
        nptr.setLinkNext(start);
        end = nptr;
    }
    size++;
}
public void insertAtPos(int val , int pos)
{
    Node nptr = new Node(val, null, null);
    if (pos == 1)
    {
        insertAtStart(val);
        return;
    }
```

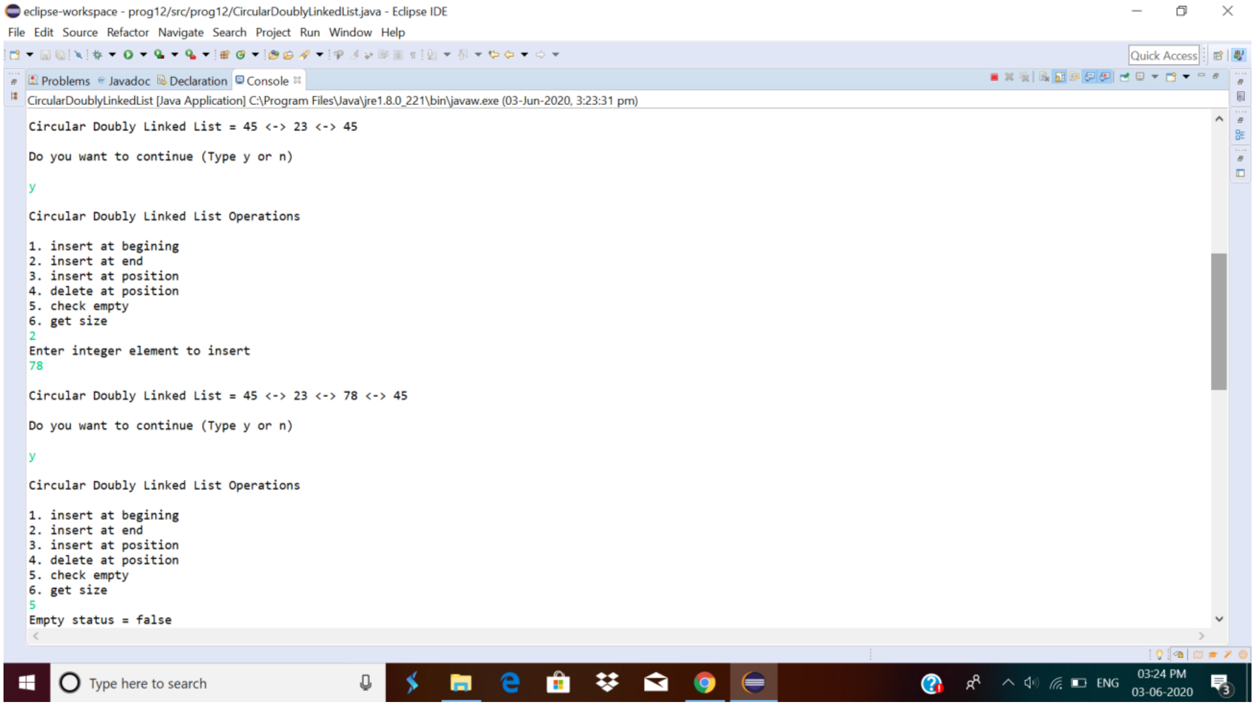
```
        Node ptr = start;
        for (int i = 2; i <= size; i++)
        {
            if (i == pos)
            {
                Node tmp = ptr.getLinkNext();
                ptr.setLinkNext(nptr);
                nptr.setLinkPrev(ptr);
                nptr.setLinkNext(tmp);
                tmp.setLinkPrev(nptr);
            }
            ptr = ptr.getLinkNext();
        }
        size++ ;
    }
    /* Function to delete node at position  */
    public void deleteAtPos(int pos)
    {
        if (pos == 1)
        {
            if (size == 1)
            {
                start = null;
                end = null;
                size = 0;
                return;
            }
            start = start.getLinkNext();
            start.setLinkPrev(end);
            end.setLinkNext(start);
            size--;
            return ;
        }
        if (pos == size)
        {
            end = end.getLinkPrev();
            end.setLinkNext(start);
            start.setLinkPrev(end);
            size-- ;
        }
        Node ptr = start.getLinkNext();
        for (int i = 2; i <= size; i++)
        {
            if (i == pos)
            {
                Node p = ptr.getLinkPrev();
                Node n = ptr.getLinkNext();

                p.setLinkNext(n);
                n.setLinkPrev(p);
                size-- ;
                return;
            }
            ptr = ptr.getLinkNext();
        }
    }
    /* Function to display status of list */
    public void display()
    {
        System.out.print("\nCircular Doubly Linked List = ");
        Node ptr = start;
        if (size == 0)
        {
            System.out.print("empty\n");
            return;
        }
        if (start.getLinkNext() == start)
        {
            System.out.print(start.getData()+ " <-> "+ptr.getData()+ "\n");
            return;
        }
        System.out.print(start.getData()+ " <-> ");
        ptr = start.getLinkNext();
        while (ptr.getLinkNext() != start)
        {
            System.out.print(ptr.getData()+ " <-> ");
            ptr = ptr.getLinkNext();
        }
        System.out.print(ptr.getData()+ " <-> ");
        ptr = ptr.getLinkNext();
        System.out.print(ptr.getData()+ "\n");
    }
}
package prog12;

import java.util.Scanner;

public class CircularDoublyLinkedList
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        linkedList list = new linkedList();
        System.out.println("Circular Doubly Linked List Test\n");
        char ch;
        do
        {
            System.out.println("\nCircular Doubly Linked List Operations\n");
            System.out.println("1. insert at begining");
            System.out.println("2. insert at end");
            System.out.println("3. insert at position");
            System.out.println("4. delete at position");
            System.out.println("5. check empty");
            System.out.println("6. get size");

            int choice = scan.nextInt();
            switch (choice)
            {
            case 1 :
                System.out.println("Enter integer element to insert");
                list.insertAtStart( scan.nextInt() );
                break;
            case 2 :
                System.out.println("Enter integer element to insert");
                list.insertAtEnd( scan.nextInt() );
                break;
            case 3 :
                System.out.println("Enter integer element to insert");
                int num = scan.nextInt() ;
                System.out.println("Enter position");
                int pos = scan.nextInt() ;
                if (pos < 1 || pos > list.getSize() )
                    System.out.println("Invalid position\n");
                else
                    list.insertAtPos(num, pos);
                break;
            case 4 :
                System.out.println("Enter position");
                int p = scan.nextInt() ;
                if (p < 1 || p > list.getSize() )
                    System.out.println("Invalid position\n");
                else
                    list.deleteAtPos(p);
                break;
            case 5 :
                System.out.println("Empty status = "+ list.isEmpty());
                break;
            case 6 :
                System.out.println("Size = "+ list.getSize() +" \n");
                break;
            default :
                System.out.println("Wrong Entry \n ");
                break;
            }
            list.display();
            System.out.println("\nDo you want to continue (Type y or n) \n");
            ch = scan.next().charAt(0);
        } while (ch == 'Y'|| ch == 'y');
    }
}
```

Output:


eclipse-workspace - prog12/src/prog12/CircularDoublyLinkedList.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
ENG
03:24 PM
03-06-2020

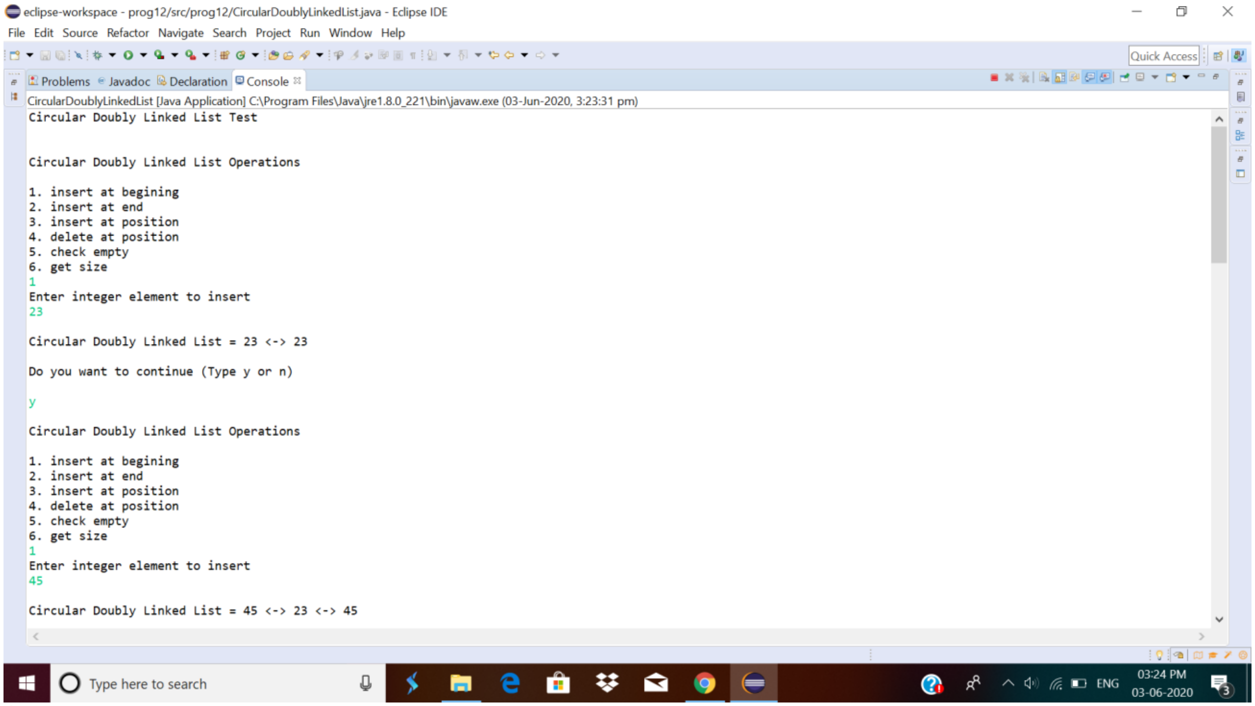
CircularDoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (03-Jun-2020, 3:23:31 pm)

```
Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
6
Size = 3


Circular Doubly Linked List = 45 <-> 23 <-> 78 <-> 45

Do you want to continue (Type y or n)

y

Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
3
Enter integer element to insert
2
Enter position
4
Invalid position


Circular Doubly Linked List = 45 <-> 23 <-> 78 <-> 45
```


eclipse-workspace - prog12/src/prog12/CircularDoublyLinkedList.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
ENG
03:24 PM
03-06-2020

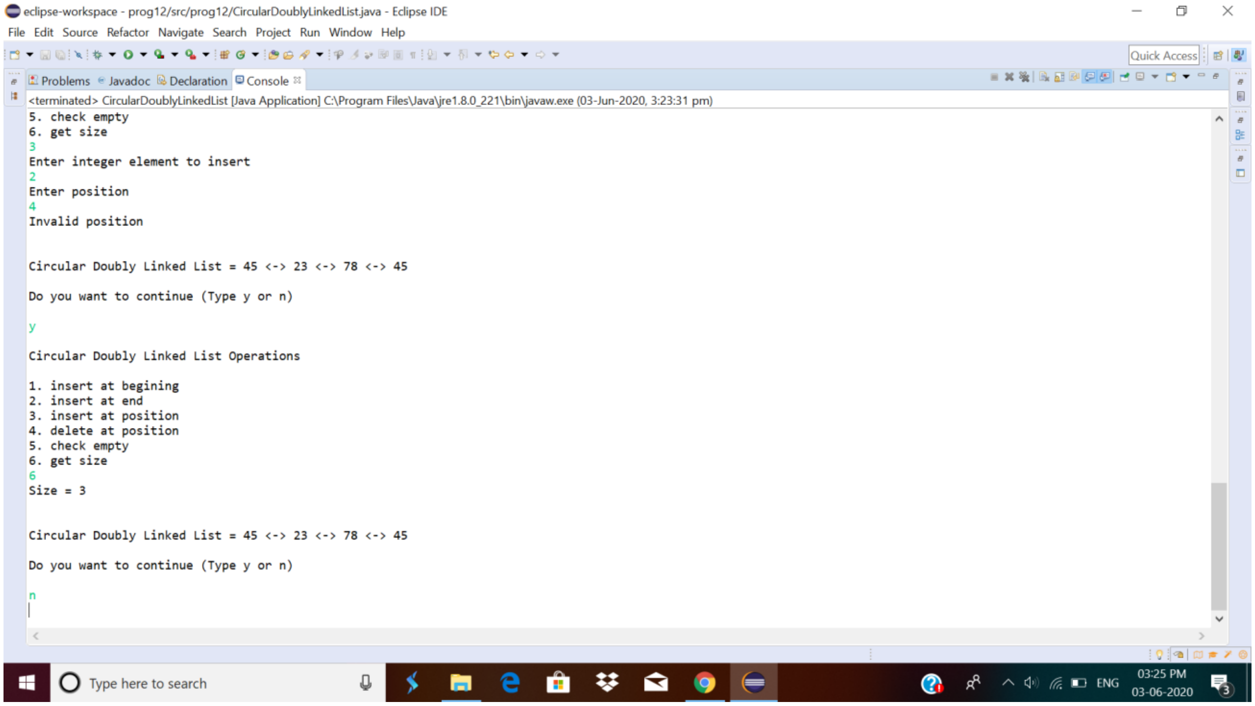
CircularDoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (03-Jun-2020, 3:23:31 pm)

```
Circular Doubly Linked List = 45 <-> 23 <-> 45

Do you want to continue (Type y or n)

y

Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
2
Enter integer element to insert
78

Circular Doubly Linked List = 45 <-> 23 <-> 78 <-> 45

Do you want to continue (Type y or n)

y

Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
5
Empty status = false
```


eclipse-workspace - prog12/src/prog12/CircularDoublyLinkedList.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
ENG
03:24 PM
03-06-2020

CircularDoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (03-Jun-2020, 3:23:31 pm)

```
Circular Doubly Linked List Test


Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
1
Enter integer element to insert
23

Circular Doubly Linked List = 23 <-> 23

Do you want to continue (Type y or n)

y

Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
1
Enter integer element to insert
45

Circular Doubly Linked List = 45 <-> 23 <-> 45
```


eclipse-workspace - prog12/src/prog12/CircularDoublyLinkedList.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
ENG
03:25 PM
03-06-2020

<terminated> CircularDoublyLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (03-Jun-2020, 3:23:31 pm)

```
5. check empty
6. get size
3
Enter integer element to insert
2
Enter position
4
Invalid position


Circular Doubly Linked List = 45 <-> 23 <-> 78 <-> 45

Do you want to continue (Type y or n)

y

Circular Doubly Linked List Operations

1. insert at begining
2. insert at end
3. insert at position
4. delete at position
5. check empty
6. get size
6
Size = 3


Circular Doubly Linked List = 45 <-> 23 <-> 78 <-> 45

Do you want to continue (Type y or n)

n
```