DAILY ONLINE ACTIVITIES SUMMARY

Date:	**02 June 2020**	**Name:**	**Asha Rudrappa Totagi**
Sem& Sec	**6thsem& A sec**	**USN:**	**4AL17CS015**

Online Test Summary

Subject	**Computer Graphics And Visualization**		
Max. Marks	**30**	**Score**	**30**

Certification Course Summary

Course	**Career Edge – Knockdown the Lockdown**		
Certificate Provider	**TCSiON**	**Duration**	**15 days**

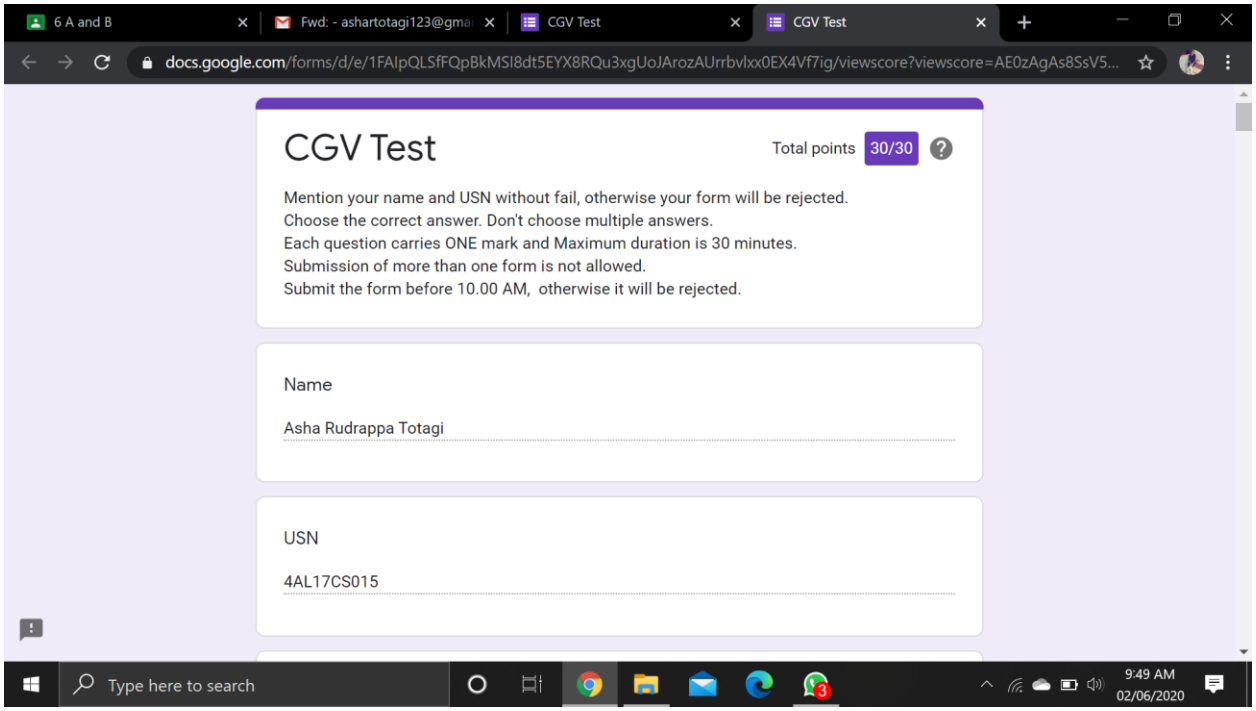
Coding Challenges

Problem Statement
Program 1:
Java program to check a loop in a linkedlist
Program 2:
Program for Inversion count of array

Status: DONE

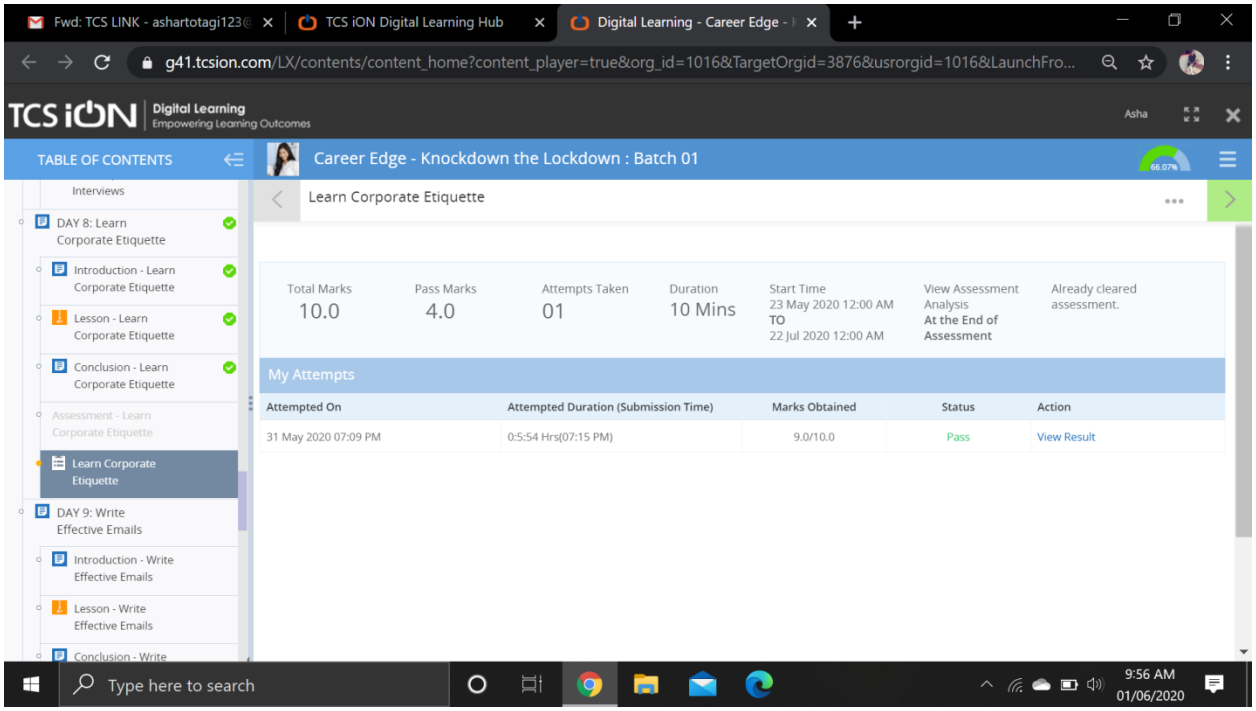
Uploaded the report in Github	**YES**
If yes Repository name	**Daily Status**
Uploaded the report in slack	**YES**

Online Test Details: (Attach the snapshot and briefly write the report for the same)



CGV IA2 test was held today i.e, 02 June 2020. Out of 30 marks I scored 30.

Certification Course Details: (Attach the snapshot and briefly write the report for the same



DAY8 (02-06-2020) – Introduction to Learn Corporate Etiquette.

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

Program 1:

```
import java.util.*;

public class LinkedList {

   static Node head; // head of list

   /* Linked list Node*/
   static class Node {
      int data;
      Node next;
      Node(int d)
      {
         data = d;
         next = null;
      }
   }

   /* Inserts a new Node at front of the list. */
   static public void push(int new_data)
   {
      /* 1 & 2: Allocate the Node &
               Put in the data*/
      Node new_node = new Node(new_data);

      /* 3. Make next of new Node as head */
      new_node.next = head;

      /* 4. Move the head to point to new Node */
      head = new_node;
   }

   // Returns true if there is a loop in linked
   // list else returns false.
   static boolean detectLoop(Node h)
   {
      HashSet<Node> s = new HashSet<Node>();
      while (h != null) {
         // If we have already has this node
         // in hashmap it means their is a cycle
```

```
            // (Because you we encountering the
            // node second time).
            if (s.contains(h))
                return true;

            // If we are seeing the node for
            // the first time, insert it in hash
            s.add(h);

            h = h.next;
        }

        return false;
    }

    /* Driver program to test above function */
    public static void main(String[] args)
    {
        LinkedList llist = new LinkedList();

        llist.push(20);
        llist.push(4);
        llist.push(15);
        llist.push(10);

        /*Create loop for testing */
        llist.head.next.next.next.next = llist.head;

        if (detectLoop(head))
            System.out.println("Loop found");
        else
            System.out.println("No Loop");
    }
}
```

Program 2:

```
import java.util.*;
public class Main
{
public static void main(String[] args) {
  Scanner sc=new Scanner(System.in);
  int t=sc.nextInt();
```

```
   while(t>0)
   {
   int n=sc.nextInt();
   int count=0;
   int a[]=new int[n];
   for (int i=0;i<n;i++)
     a[i]=sc.nextInt();
   for(int i=0;i<n;i++)
    for(int j=i+1;j<n-1;j++)
    {
        if(a[i]>a[j])
        {
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
            count++;
        }
    }

System.out.println(count);
t--;
   }
}
}
```