DAILY ONLINE ACTIVITIES SUMMARY

Date:	**04 June 2020**	**Name:**	**Asha Rudrappa Totagi**
Sem& Sec	**6thsem& A sec**	**USN:**	**4AL17CS015**

Online Test Summary

Subject	**Operating System**		
Max. Marks	**30**	**Score**	**22**

Certification Course Summary

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

Coding Challenges

Problem Statement

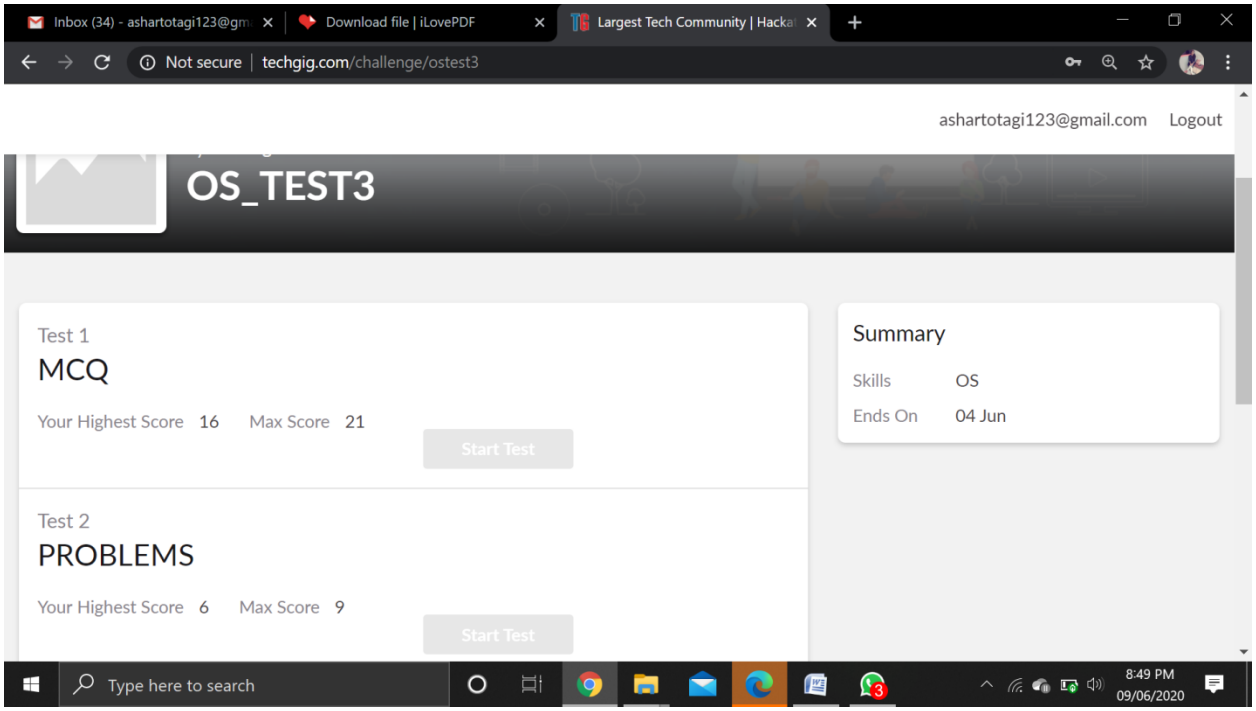
Program 1: Take two strings, return a string of the form short+long+short, with the shorter string on the outsides and the longer string on the inside. The strings will not be the same length, but they may be empty (length 0).

Program 2: Write a Java program to implement Queue Using Array And Class

Status: DONE

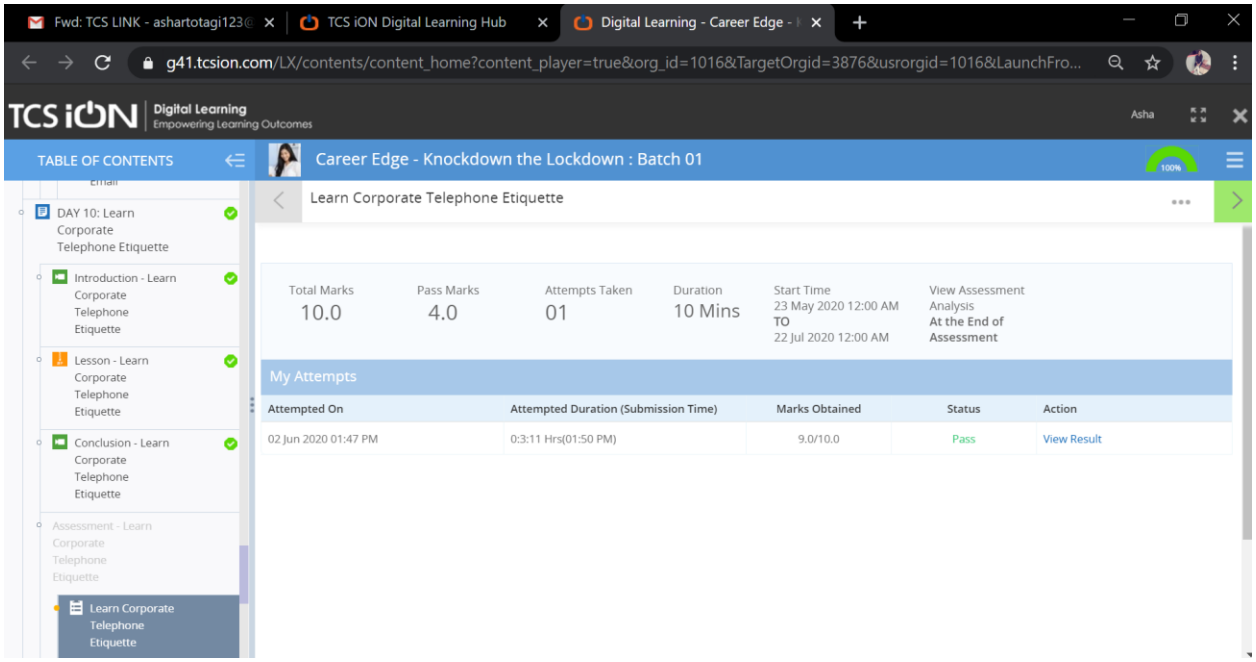
Uploaded the report in Github	**YES**
If yes Repository name	**Daily Status**
Uploaded the report in slack	**YES**

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



OS IA3 test was held today i.e, 04 June 2020. Out of 30 marks I scored 22.

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



DAY10 (04-06-2020) – Introduction to Learn Corporate Telephone Etiquette.

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

Program 1:

```
str1=input("Str1=")
str2=input("Str2=")
count1=0
count2=0
for i in str1:
   count1=count1+1
for j in str2:
   count2=count2+1
if count1<count2:
   print("Combined_str("+str1+","+str2+")"+"="+(str1+str2+str1))
else:
   print("Combined_str("+str1+","+str2+")"+"="+(str2+str1+str2))
```

Program 2:

```
import java.util.*;

class arrayQueue

{

protected int Queue[] ;

protected int front, rear, size, len;

public arrayQueue(int n)

{

size = n;

len = 0;

Queue = new int[size];

front = -1;

rear = -1;

}
```

```
public boolean isEmpty()
{
return front == -1;
}
public boolean isFull()
{
return front==0 &amp;amp;&amp;amp; rear == size -1 ;
}
public int getSize()
{
return len ;
}

public int peek()
{
if (isEmpty())
throw new NoSuchElementException(&amp;quot;Underflow Exception&amp;quot;);
return Queue[front];
}
public void insert(int i)
{
if (rear == -1)
{
front = 0;
rear = 0;
```

```
Queue[rear] = i;
}
else if (rear + 1 &gt;= size)
throw new IndexOutOfBoundsException(&quot;Overflow Exception&quot;);
else if ( rear + 1 &lt; size)
Queue[++rear] = i;
len++ ;
}
public int remove()
{
if (isEmpty())
throw new NoSuchElementException(&quot;Underflow Exception&quot;);
else
{
len-- ;

int ele = Queue[front];
if ( front == rear)
{
front = -1;
rear = -1;
}
else
front++;
return ele;
```

```
}
}
public void display()
{
System.out.print(&quot;\nQueue = &quot;);
if (len == 0)
{
System.out.print(&quot;Empty\n&quot;);
return ;
}
for (int i = front; i &lt;= rear; i++)
System.out.print(Queue[i]+&quot; &quot;);
System.out.println();
}
}
public class Main
{
public static void main(String[] args)
{
Scanner scan = new Scanner(System.in);
System.out.println(&quot;Array Queue Test\n&quot;);
System.out.println(&quot;Enter Size of Integer Queue &quot;);
int n = scan.nextInt();
arrayQueue q = new arrayQueue(n);
char ch;
```

```
do{
System.out.println(&quot;\nQueue Operations&quot;);
System.out.println(&quot;1. insert&quot;);
System.out.println(&quot;2. remove&quot;);
System.out.println(&quot;3. peek&quot;);
System.out.println(&quot;4. check empty&quot;);
System.out.println(&quot;5. check full&quot;);
System.out.println(&quot;6. size&quot;);
int choice = scan.nextInt();
switch (choice)
{
case 1 :
System.out.println(&quot;Enter integer element to insert&quot;);
try
{
q.insert( scan.nextInt() );
}
catch(Exception e)
{
System.out.println(&quot;Error : &quot; +e.getMessage());
}
break;
case 2 :
try
{
```

```
System.out.println(&quot;Removed Element = &quot;+q.remove());
}
catch(Exception e)
{
System.out.println(&quot;Error : &quot; +e.getMessage());
}
break;
case 3 :
try
{
System.out.println(&quot;Peek Element = &quot;+q.peek());
}
catch(Exception e)
{
System.out.println(&quot;Error : &quot;+e.getMessage());
}
break;
case 4 :
System.out.println(&quot;Empty status = &quot;+q.isEmpty());
break;
case 5 :

System.out.println(&quot;Full status = &quot;+q.isFull());
break;
case 6 :
```

System.out.println("Size = "+ q.getSize());

break;

default : System.out.println("Wrong Entry \n ");

break;

}

q.display();

System.out.println("\nDo you want to continue (Type y or n) \n");

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

} while (ch == 'Y'|| ch == 'y');

}

}