DAILY ONLINE ACTIVITIES
SUMMARY

Date:	**18/05/2020**	**Name:**	**SUMAN RATHOD**
Sem & Sec	**6th sem & B sec**	**USN:**	**4AL17CS115**
Online Test Summary			
Subject	**CNSC IA Test**		
Max. Marks	**60**	**Score**	**47**
Certification Course Summary			
Course	**Introduction to Full Stack Development**		
Certificate Provider	**Great Learning**	**Duration**	**1.5 hr(spent by me on that day to learn)**
Coding Challenges			

Problem Statement:

1. Java code to find shortest palindrome for the given string.

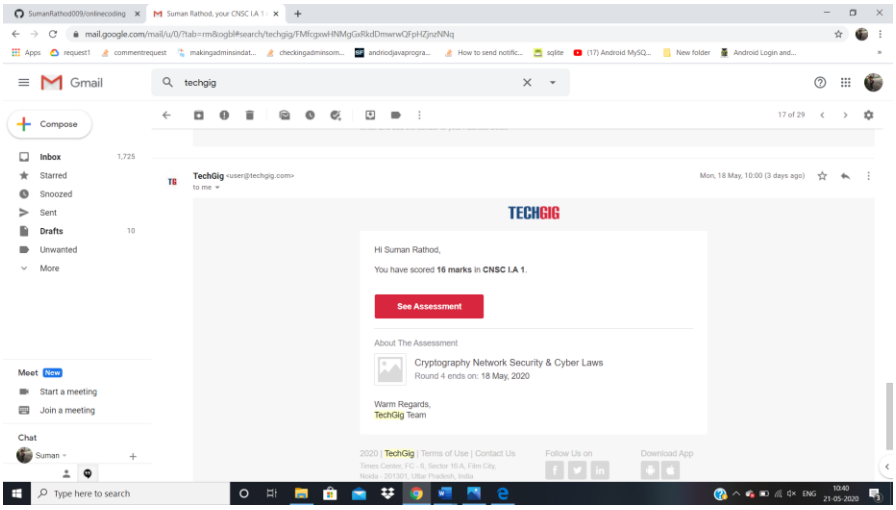
2. Write a simple code to identify given linked list is palindrome or not by using stack. First take a Stack. Traverse through each node of the linked list and push each node value to Stack.
.

Status: Completed	
Uploaded the report in Github	**Yes**
If yes Repository name	https://github.com/SumanRathod009/onlinecoding

Uploaded the report in slack	**Yes**

Online Test Details

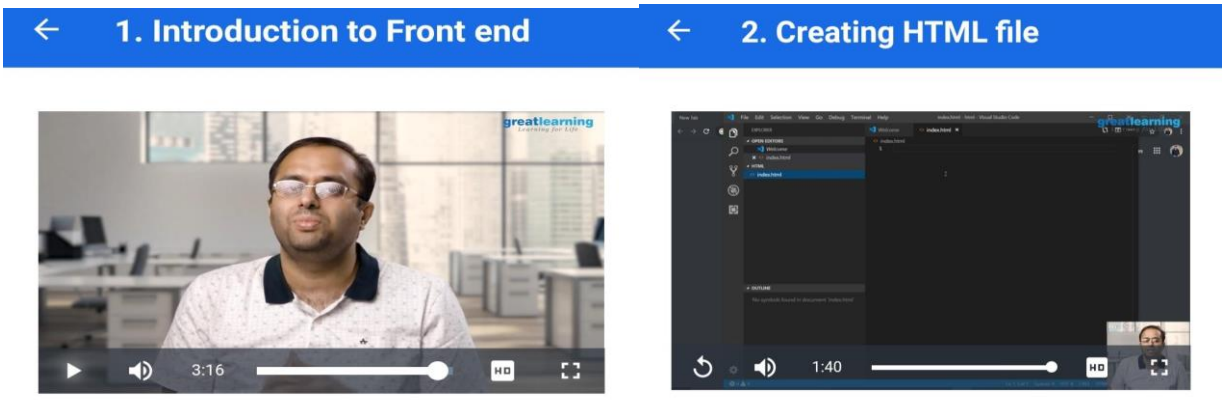
CGV TEST Details:



Online Certification Details

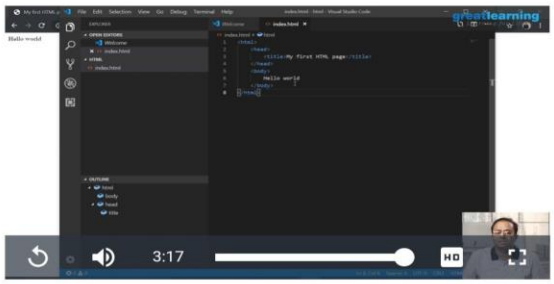
Lessons completed:

1. Introduction to front end
2. Creating HTML file
3. HTML Structure
4. Paragraph Tags



5. More on Head Tags

3. HTML Structure



Assignment

HTML Structure

Submission type : Online upload

Thank you for your submission, we're grading your assignment now.

Submitted Assignment

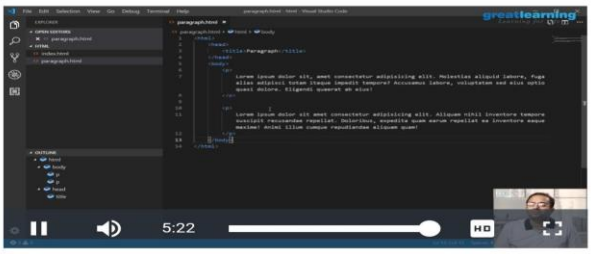
Resubmit Assignment

Screenshot_20200518-185354_Dcoder.jpg

Screenshot_20200518-185111_Dcoder.jpg

Submitted at Mon, May 18

4. Paragraph Tags



Assignment

Paragraph Tags

Submission type : Online upload

Thank you for your submission, we're grading your assignment now.

Submitted Assignment

Resubmit Assignment

Screenshot_20200518-191926_Dcoder.jpg

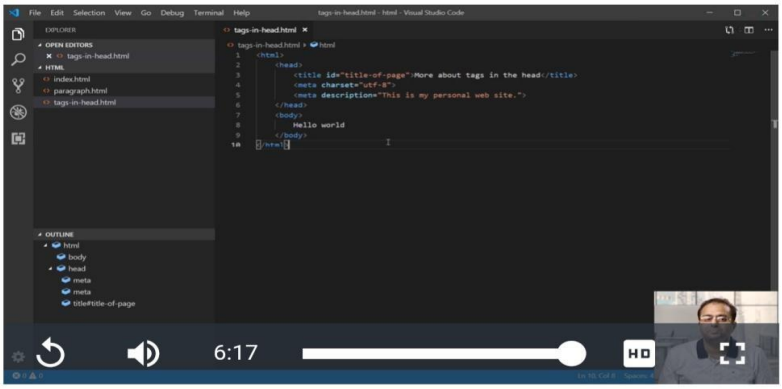
Screenshot_20200518-191419_Dcoder.jpg

Screenshot_20200518-191814_Dcoder.jpg

Screenshot_20200518-191757_Dcoder.jpg

Submitted at Mon, May 18

5. More on Head Tags



Coding Challenge Details

1.Using methods charAt() & length() of String class, write a program to print the frequency of each character in a string.

```
1  import java.util.*;
2
3  public class Main
4  {
5  public static void main(String args[])
6  {
7  int i;
8  String s;
9  int c[] = new int[256];
10 Scanner sc = new Scanner(System.in);
11 System.out.print("Enter a String : ");
12 s=sc.nextLine();
13 for (i = 0; i < s.length(); i++)
14 c[(int) s.charAt(i)]++;
15 for (i = 0; i < 256; i++) {
16 if (c[i] != 0) {
17 System.out.println((char)i + ": " + c[i]);
18 }
19 }
20 }
21 }
```

```
✕ Terminal
Enter a String : amrutha
a: 2
h: 1
m: 1
r: 1
t: 1
u: 1

Process finished.
```

2. Write down a java program to print even and odd numbers series respectively from two threads: t1 and t2 synchronizing on a shared object

Let t1 print message “ping — >” and t2 print message “,—-pong”.

```
class OddThread extends Thread
{
int limit;
sharedPrinter printer;
public OddThread(int limit, sharedPrinter printer)
{
this.limit = limit;
this.printer = printer;
}
@Override
public void run()
{
int oddNumber = 1;
while (oddNumber <= limit)
{
printer.printOdd(oddNumber);
oddNumber = oddNumber + 2;
}
}
}

class EvenThread extends Thread
{
int limit;
sharedPrinter printer;
public EvenThread(int limit, sharedPrinter printer
{
this.limit = limit;
this.printer = printer;
}
@Override
public void run()
{
int evenNumber = 2;
while (evenNumber <= limit)
{
printer.printEven(evenNumber);
evenNumber = evenNumber + 2;
}
}
}
class sharedPrinter
{

boolean isOddPrinted = false;

synchronized void printOdd(int number)
{
while (isOddPrinted)
{
try
{
wait();
}
catch (InterruptedException e)
{
e.printStackTrace();
}
}
System.out.println(Thread.currentThread().getName(
isOddPrinted = true;
try
{
Thread.sleep(1000);
}
catch (InterruptedException e)
{
e.printStackTrace();
}
notify();
}

synchronized void printEven(int number)
{
while (! isOddPrinted)
{
try
{
wait();
}
```

```
82  catch (InterruptedException e)
83  {
84  e.printStackTrace();
85  }
86  }
87  System.out.println(Thread.currentThread().getName(
88  isOddPrinted = false;
89  try
90  {
91  Thread.sleep(1000);
92  }
93  catch (InterruptedException e)
94  {
95  e.printStackTrace();
96  }
97  notify();
98  }
99  }
100 public class Main
101 {
102 public static void main(String[] args)
103 {
104 sharedPrinter printer = new sharedPrinter();
105 OddThread oddThread = new OddThread(20, printer);
106 oddThread.setName("--pong");
107 EvenThread evenThread = new EvenThread(20, printer
108 evenThread.setName("ping - >");
109 oddThread.start();
110 evenThread.start();
111 }
112 }
```

```
Terminal
--pong 1
ping - > 2
--pong 3
ping - > 4
--pong 5
ping - > 6
--pong 7
ping - > 8
--pong 9
ping - > 10
--pong 11
ping - > 12
--pong 13
ping - > 14
--pong 15
ping - > 16
--pong 17
ping - > 18
--pong 19
ping - > 20

Process finished.
```