DAILY ONLINE ACTIVITIES SUMMARY

Date:	**19th May 2020**	**Name:**	**Sangeetha N A**
Sem & Sec	**8th Semester 'B' Section**	**USN:**	**4AL16CS083**

Online Test Summary

Subject	**Big Data Analytics**		
Max. Marks	**30**	**Score**	**21**

Certification Course Summary

Course	**TCS ION**		
Certificate Provider	**TCS ION**	**Duration**	**3 hours**

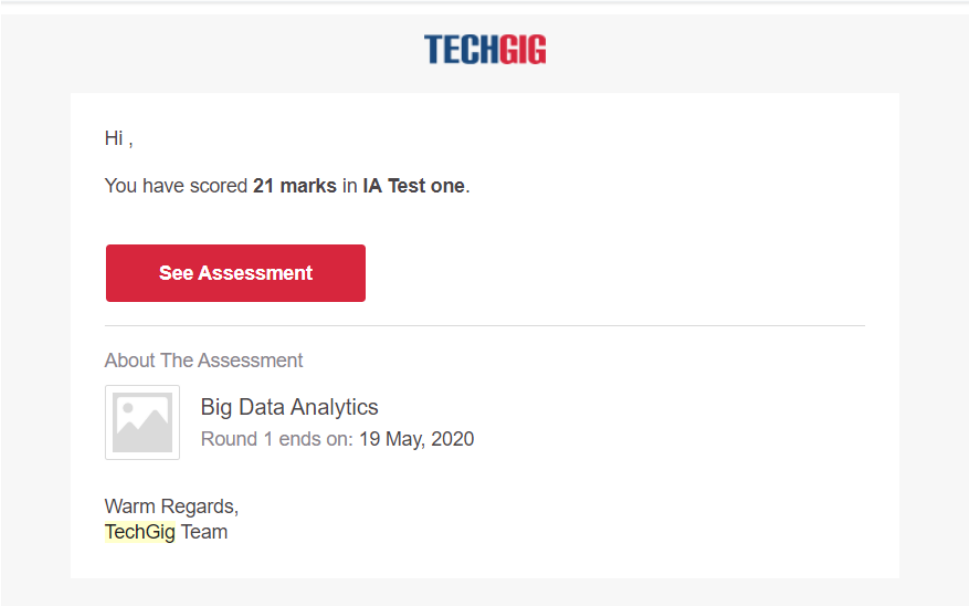
Coding Challenges

Problem Statement: 1. find out what will be the shortest palindrome string by using simple java program.
2. Write a simple code to identify given linked list is palindrome or not by using stack.

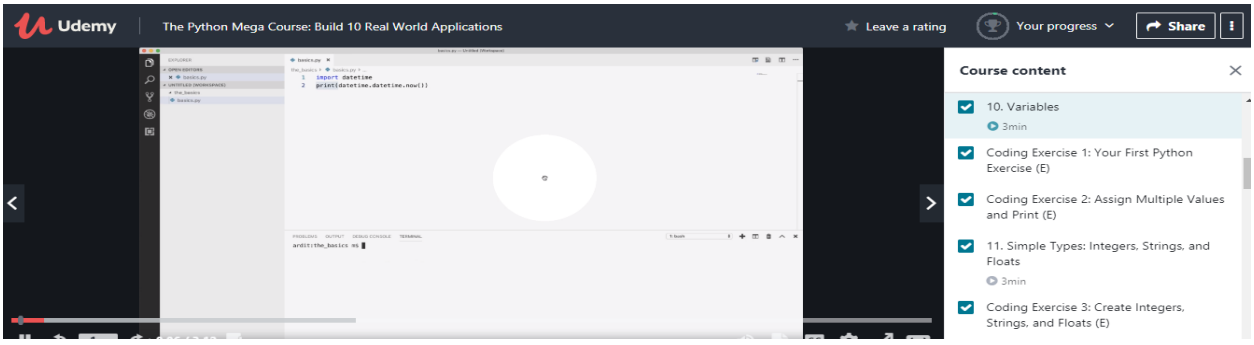
Status: completed

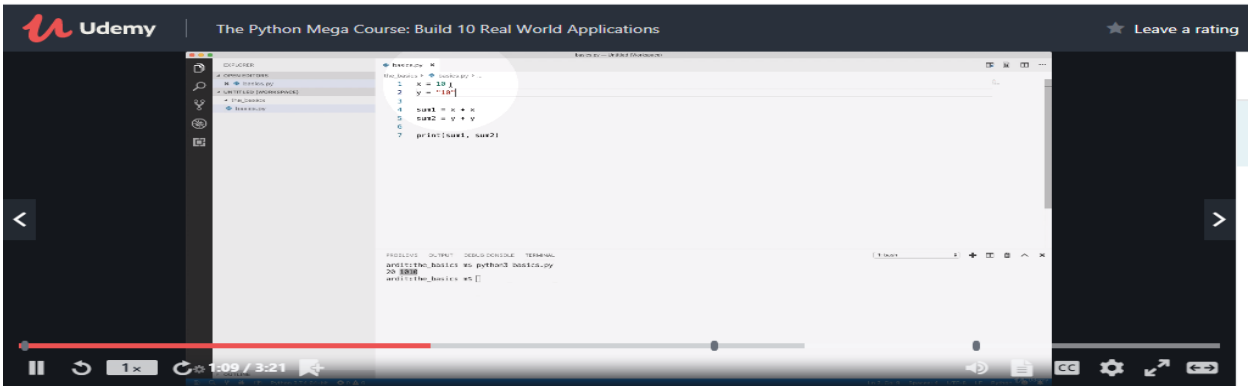
Uploaded the report in Github	**yes**
If yes Repository name	**sangeethana**
Uploaded the report in slack	**yes**

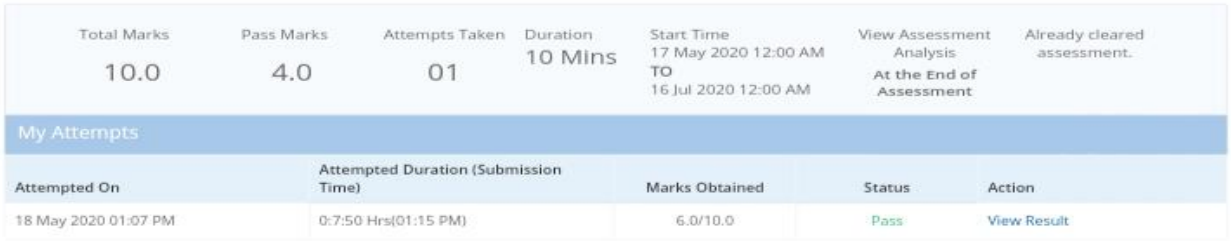
Online Test Details: (Attach the snapshot and briefly write the report for the same)



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







Total Marks	Pass Marks	Attempts Taken	Duration	Start Time	View Assessment Analysis	Already cleared assessment.
10.0	4.0	01	10 Mins	17 May 2020 12:00 AM TO 16 Jul 2020 12:00 AM	At the End of Assessment	

My Attempts

Attempted On	Attempted Duration (Submission Time)	Marks Obtained	Status	Action
18 May 2020 01:07 PM	0:7:50 Hrs(01:15 PM)	6.0/10.0	Pass	View Result

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

Program 1:

```
package shortestpalindromeexample.java;
import java.util.Scanner;

public class ShortestPalindromeDemo {

public static String shortestPalindrome(String str) {

int x=0;
int y=str.length()-1;

while(y>=0){
if(str.charAt(x)==str.charAt(y)){
x++;
}
y--;
}

if(x==str.length())
return str;

String suffix = str.substring(x);
String prefix = new StringBuilder(suffix).reverse().toString();
String mid = shortestPalindrome(str.substring(0, x));

return prefix+mid+suffix;
}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Enter a String to find out shortest palindrome");
```

```
String str=in.nextLine();

System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));

}
```

Program 2:

```
import java.util.Stack;

// Data Structure to store a linked list node
class Node {
int data;
Node next;

Node(int i)
{
	this.data = i;
	this.next = null;
}

};

class Main
{
// Function to determine if a given linked list is palindrome or not
public static boolean isPalindrome(Node head)
{
// construct an empty stack
Stack s = new Stack<>();

	// push all elements of the linked list into the stack
	Node node = head;
	while (node != null) {
		s.push(node.data);
		node = node.next;
	}

	// traverse the linked list again
	node = head;
	while (node != null)
	{
		// pop the top element from the stack
		int top = s.pop();
```

```
                // compare the popped element with current node's data
                // return false if mismatch happens
                if (top != node.data) {
                        return false;
                }

                // advance to the next node
                node = node.next;
        }

        // we reach here only when the linked list is palindrome
        return true;
}

public static void main(String[] args)
{
        Node head = new Node(1);
        head.next = new Node(2);
        head.next.next = new Node(3);
        head.next.next.next = new Node(2);
        head.next.next.next.next = new Node(1);

        if (isPalindrome(head)) {
                System.out.print("Linked List is a palindrome.");
        } else {
                System.out.print("Linked List is not a palindrome.");
        }
}

}
```