DAILY ONLINE ACTIVITIES SUMMARY

Date:	16-06-2020	Name:	M.C Suchithra Heggade
Sem & Sec	6th Sem 'A' Sec	USN:	4AL17CS047

Online Test Summary			
Subject	Python		
Max. Marks	20	Score	15

Certification Course Summary			
Course	Front end Development-HTML		
Certificate Provider	Great Learning	Duration	5 hr

Coding Challenges

BST

Write a Python program to check whether a given a binary tree is a valid binary search tree (BST) or not?

SLL

Write a Java Program to Find the Middle Node of a Linked list in a Singlepass.

Status: Completed

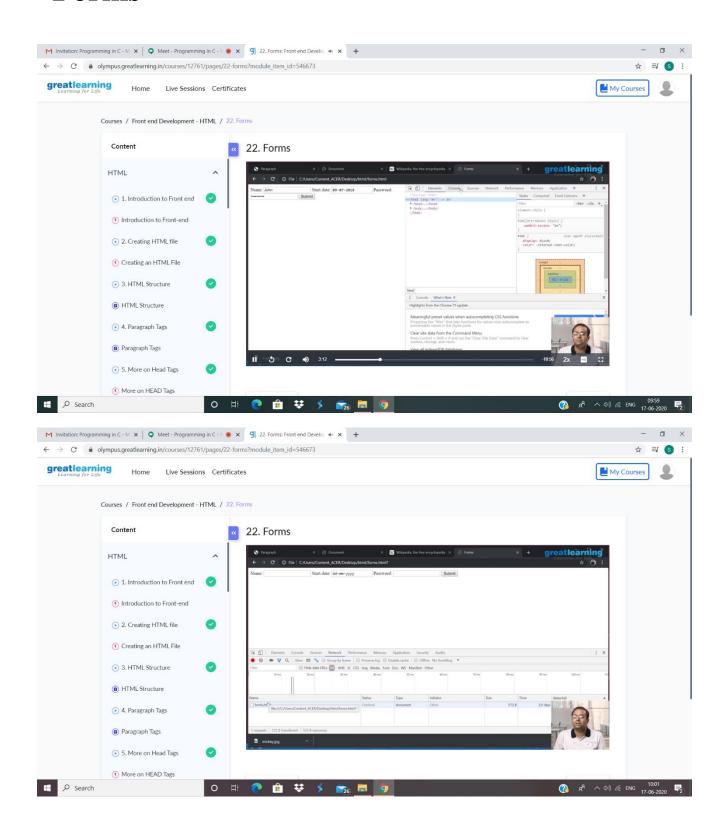| Uploaded the report in Github	yes
If yes Repository name	https://github.com/Suchitraheggade/certification-and-Online-coding https://github.com/suchitraheggade/Workshop-on-Python-Program
Uploaded the report in slack	yes

Online Test Details:

Python:

Certification Course Details:

Topics completed:

Forms

Coding Challenges Details:

1.BST

Write a Python program to check whether a given a binary tree is a valid binary search tree (BST) or not?

```python
            root = root.left
        root = stack.pop()
        if prev and root.val <= prev.val:
            return False
        prev = root
        root = root.right
    return True

root = TreeNode(2)
root.left = TreeNode(1)
root.right = TreeNode(3)

result = is_BST(root)
print(result)

root = TreeNode(1)
root.left = TreeNode(2)
root.right = TreeNode(3)

result = is_BST(root)
print(result)
```

```
True
False
```

2.SLL

Write a Java Program to Find the Middle Node of a Linked list in a Singlepass.

```java
public class LinkedListTest {

    public static void main(String args[]) {
        //creating LinkedList with 5 elements including head
        LinkedList linkedList = new LinkedList();
        LinkedList.Node head = linkedList.head();
        linkedList.add( new LinkedList.Node("1"));
        linkedList.add( new LinkedList.Node("2"));
        linkedList.add( new LinkedList.Node("3"));
        linkedList.add( new LinkedList.Node("4"));

        //finding middle element of LinkedList in single pass
        LinkedList.Node current = head;
        int length = 0;
        LinkedList.Node middle = head;

        while(current.next() != null){
            length++;
            if(length%2 ==0){
                middle = middle.next();
            }
            current = current.next();
        }

        if(length%2 == 1){
            middle = middle.next();
        }

        System.out.println("length of LinkedList: " + length);
        System.out.println("middle element of LinkedList : "       + middle);

    }

}

class LinkedList{
    private Node head;
    private Node tail;
```

Execute Mode, Version, Inputs & Arguments

Stdin Inputs

JDK 11.0.4

Interactive

CommandLine Arguments

Execute

Result
CPU Time: 0.22 sec(s), Memory: 35836 kilobyte(s)   compiled and executed in 0.959 sec(s)

```
length of LinkedList: 4
middle element of LinkedList : 2
```