# **Format for uploading details in GitHub and Slack in word file format**

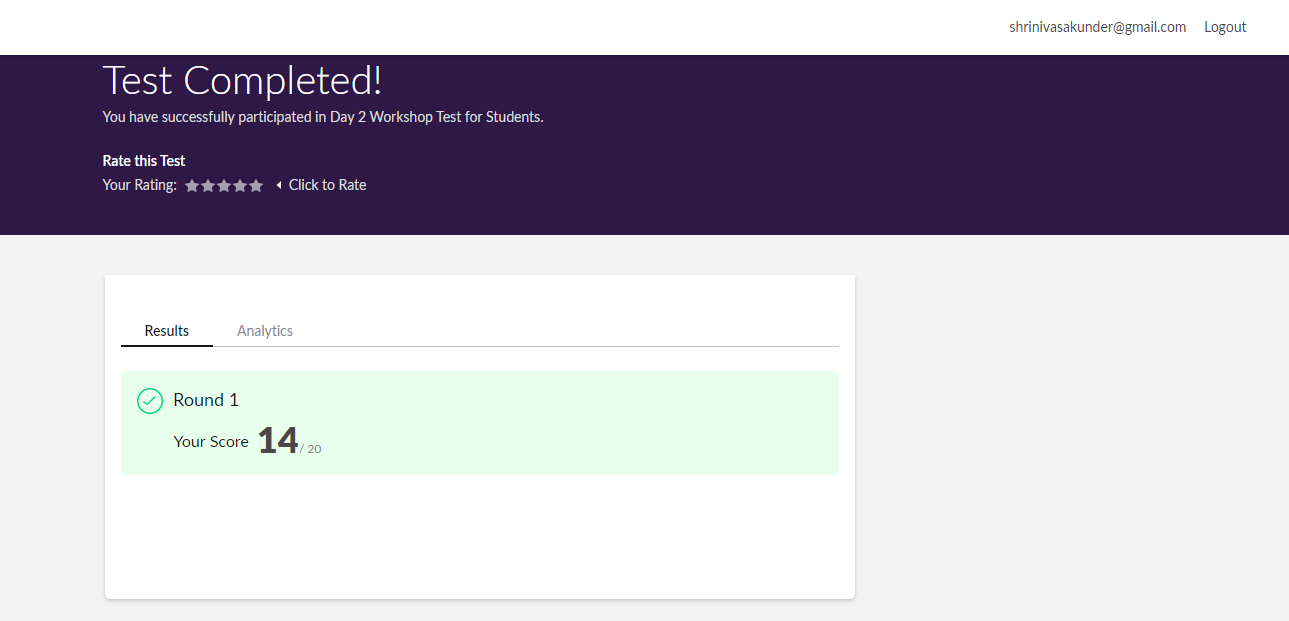
**Student Name: Shrinivasa**

**Class and Sec: VI B**

**USN: 4AL17CS092**

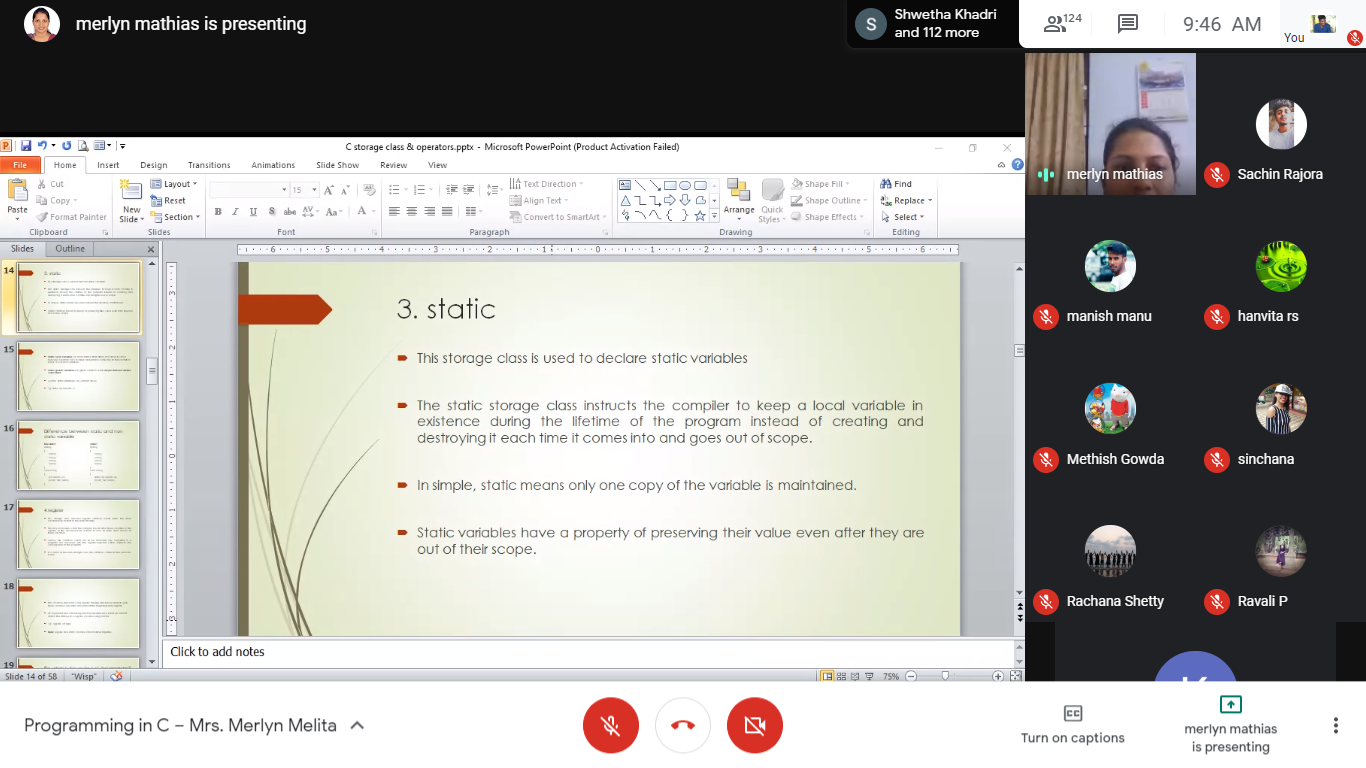
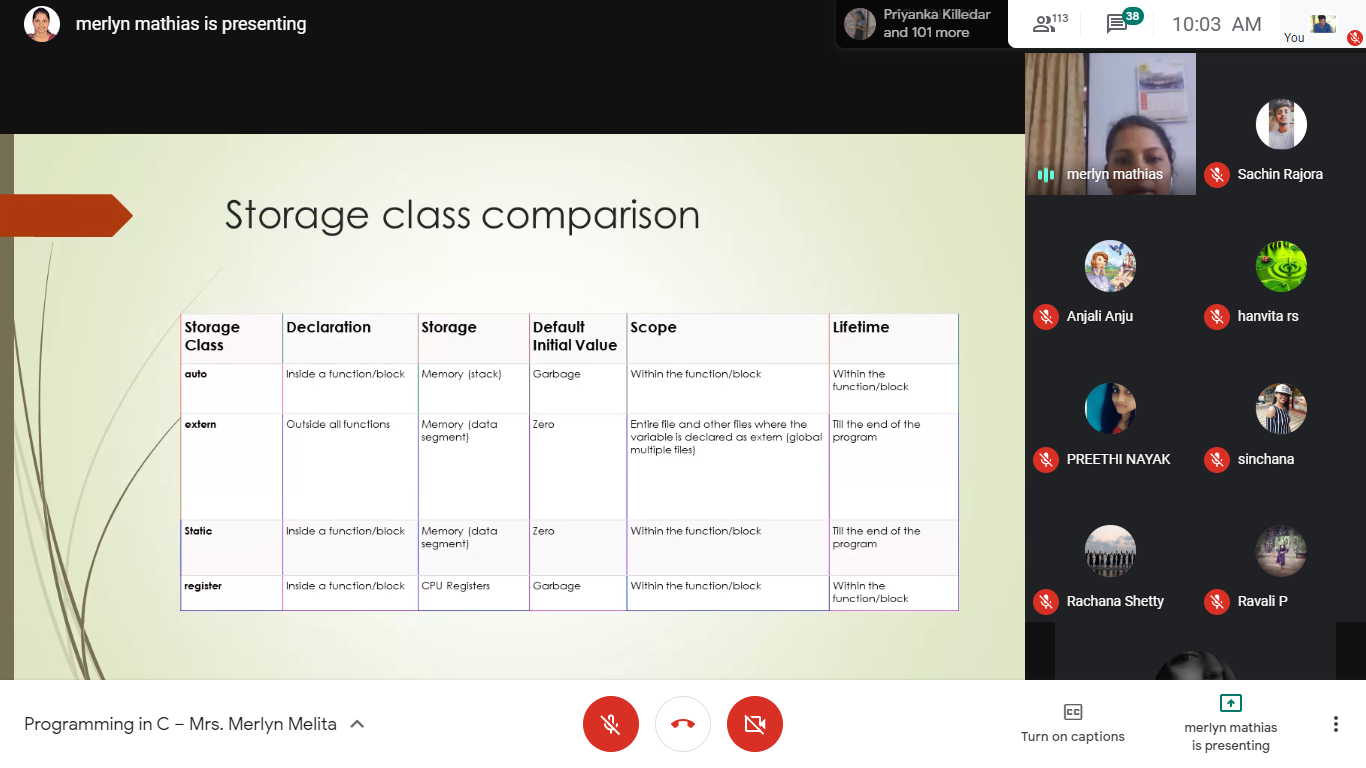
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Test Details** | | | | |
| **Subject** | **Python (Workshop quiz)** | | | |
| **Semester** | **VI - B** | | **Duration** | **30 Minutes** |
| **70%** | | **14/20** | | |

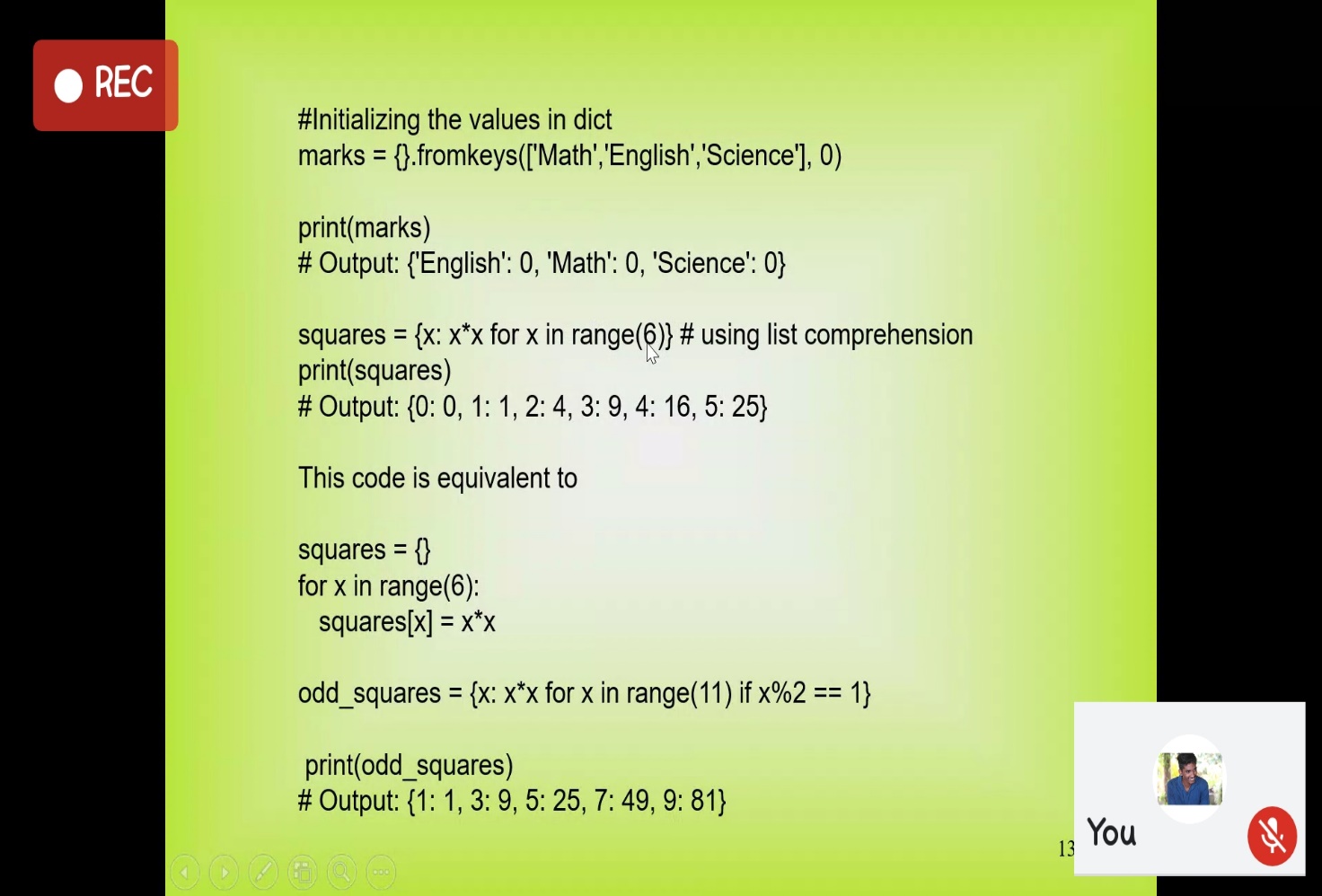
**Encl: snapshot of the test result**

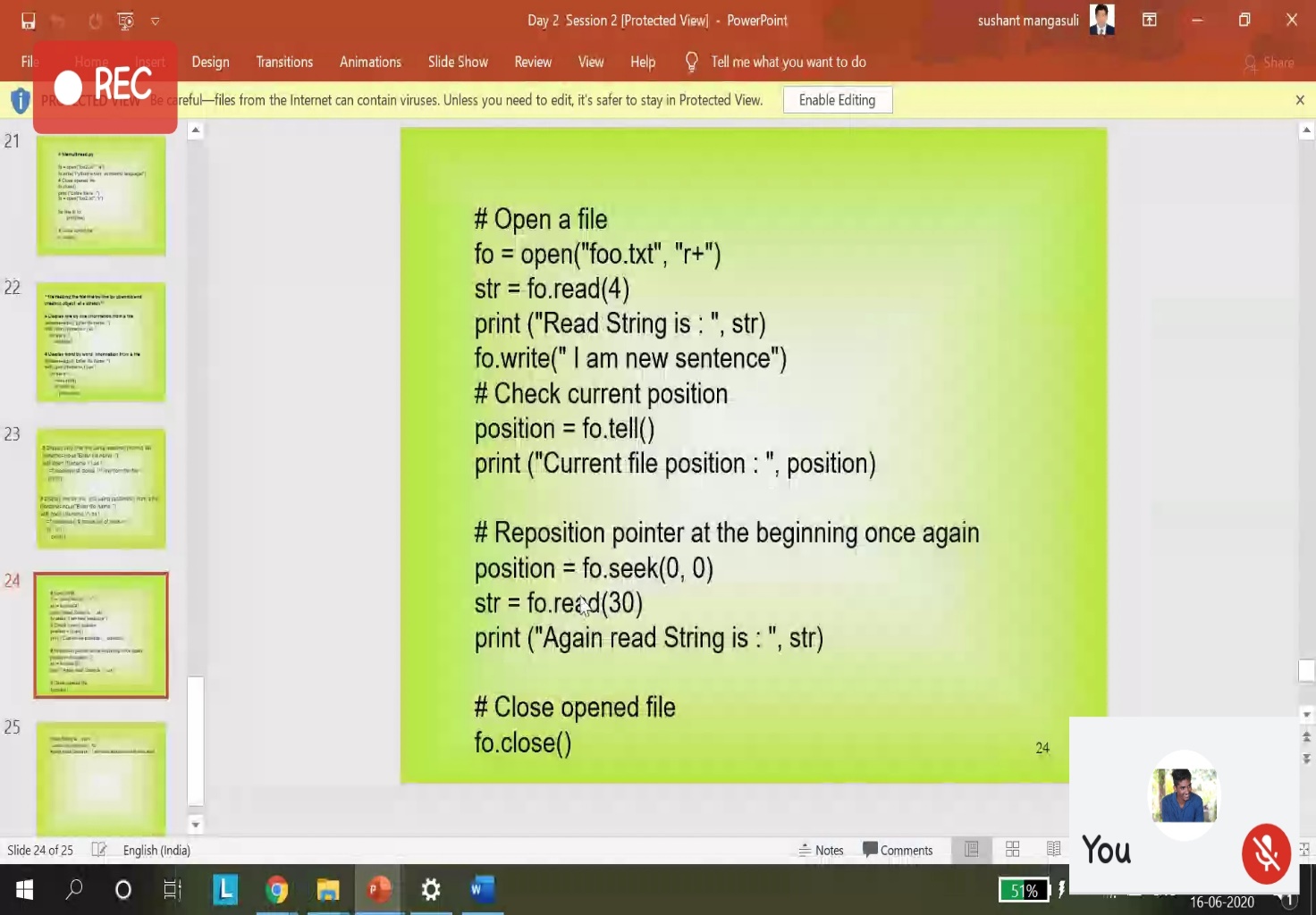


|  |  |  |  |
| --- | --- | --- | --- |
| **Pre-Placement Training Summary** | | | |
| **Pre placement training** | **9:00 am to 11:00 am - Programming in C**  **11:00 am to 1:00pm - Applications of python in DA and ML** | | |
| **Faculty** | **Mrs. Merlyn mathias,**  **Dr. Mohideen Badusha** | **Duration** | **4 hours** |

**Encl: snapshots of the daily class activities (at least two snap shots)**





|  |  |
| --- | --- |
| **Coding Challenges** | |
| **Problem Statement:1.Python program to check given tree is BST or NOT**  **2. Examples and Exercises on python. (Linked Google-colab to GitHub)**  [**https://github.com/Shrinivasakunder/certification-and-online-coding**](https://github.com/Shrinivasakunder/certification-and-online-coding) | |
| **Status: Completed** | |
| **Uploaded the report both in GitHub & Slack** | **Yes** |

**Encl: snapshots of your response to challenge.**

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

class TreeNode(object):

    def \_\_init\_\_(self, x):

        self.val = x

        self.left = None

        self.right = None

def is\_BST(root):

    stack = []

    prev = None

    while root or stack:

        while root:

            stack.append(root)

            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)

**Output:**

