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

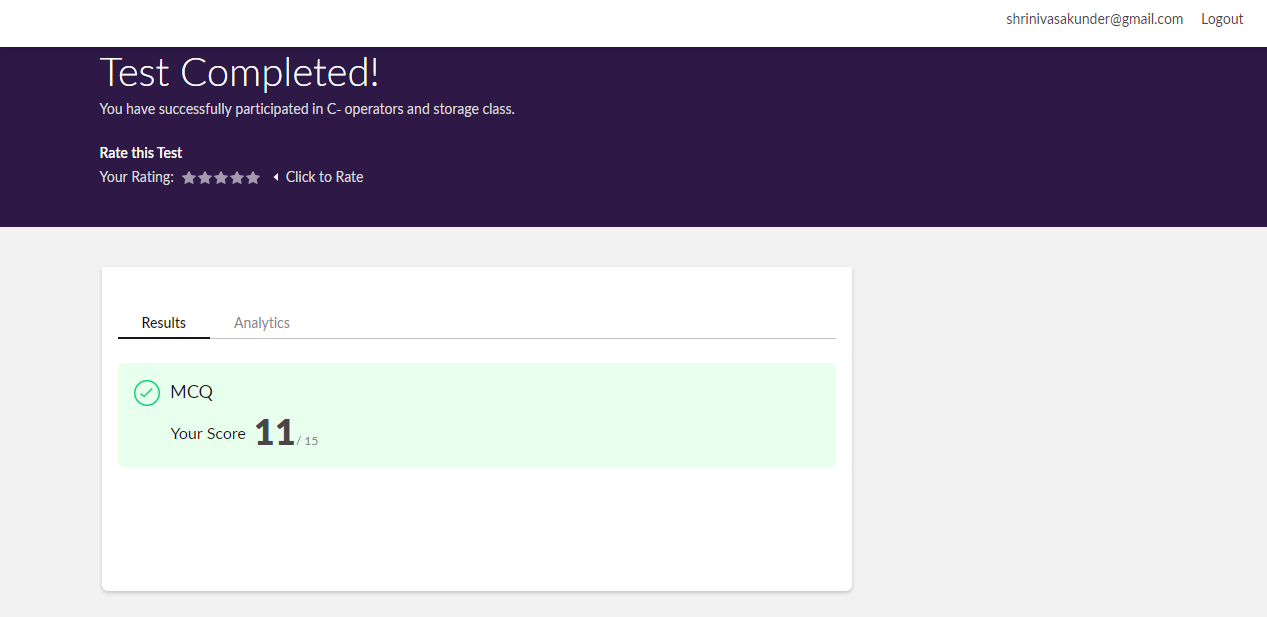
**Student Name: Shrinivasa**

**Class and Sec: VI B**

**USN: 4AL17CS092**

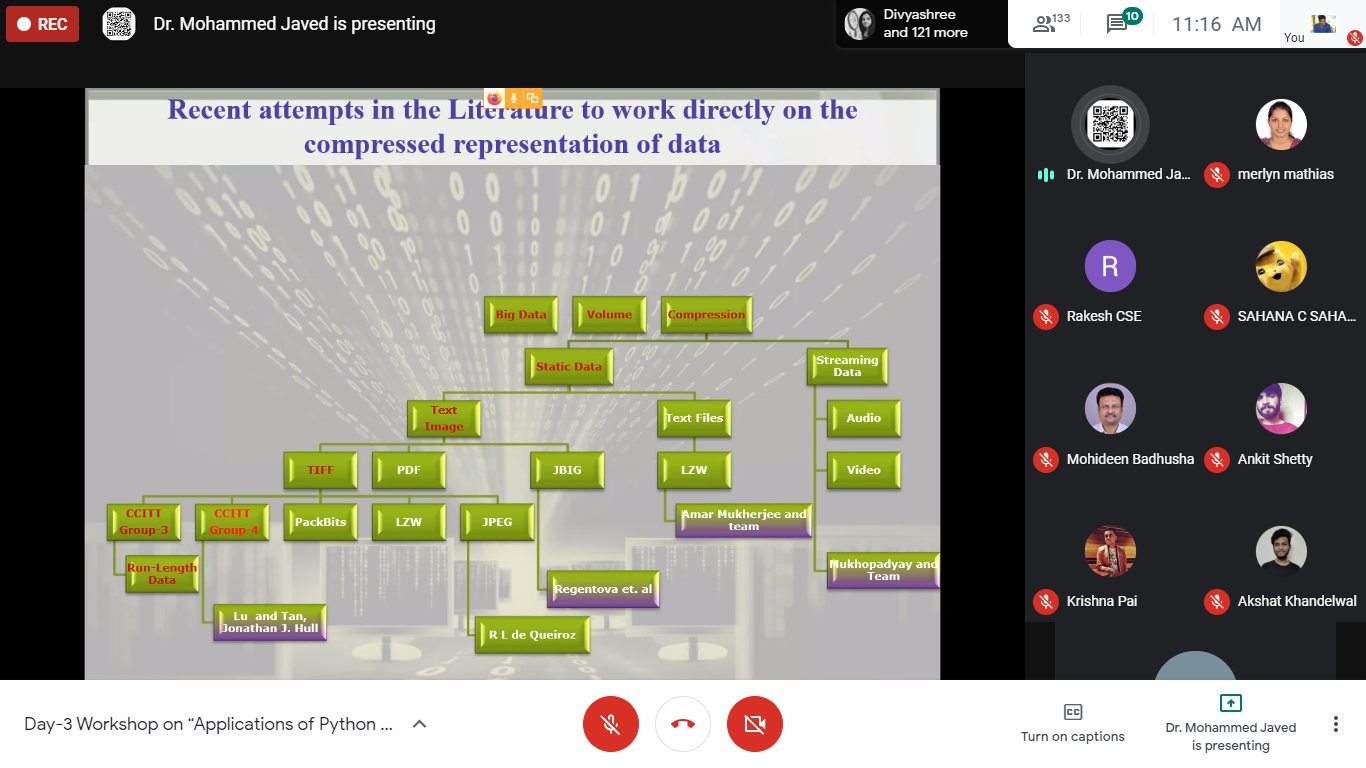
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Test Details** | | | | |
| **Subject** | **C-operators and storage classes** | | | |
| **Semester** | **VI - B** | | **Duration** | **30 Minutes** |
| **73%** | | **11/15** | | |

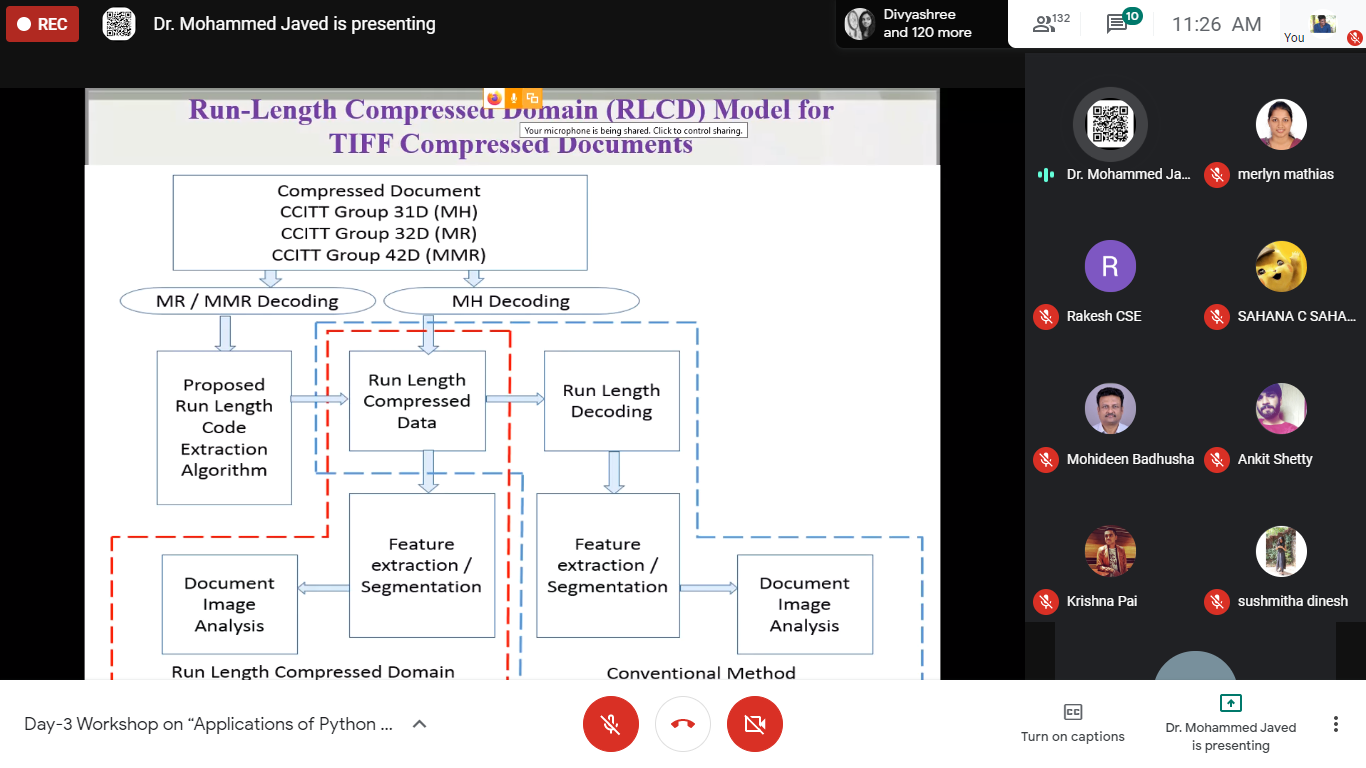
**Encl: snapshot of the test result**

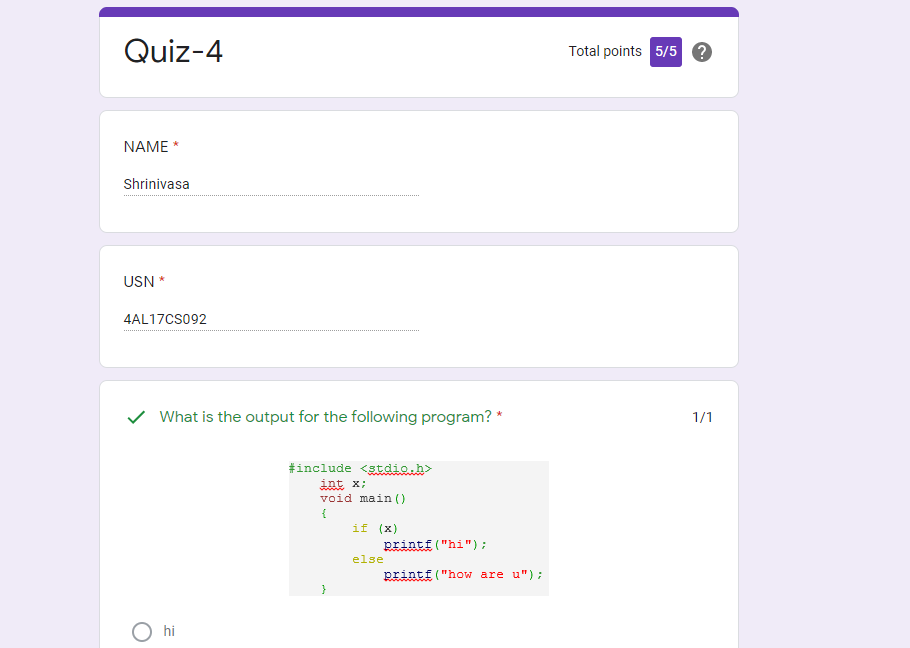


|  |  |  |  |
| --- | --- | --- | --- |
| **Pre-Placement Training Summary** | | | |
| **Pre placement training** | **9:00 am to 11:00 am - Programming in C**  **11:00 am to 1:00pm - Applications of python in DA and ML** | | |
| **Faculty** | **Mrs. Vivek Sharma,**  **Dr. Mohideen Badusha** | **Duration**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

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







|  |  |
| --- | --- |
| **Coding Challenges** | |
| **Problem Statement:1. Examples and Exercises on python. (Linked Google-colab to GitHub)**  **2.Pro1(java)**  [**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.**

**Java program to find the row, column position of a specified number (row, column position) in a given 2-dimensional array**

import java.util.\*;

public class abc {

public static void main(String[] args) {

int nums[][] = {{12, 20, 30, 40},

{15, 25, 35, 45},

{24, 29, 39, 51},

{35, 30, 39, 50},

{50, 60, 75, 72}};

int rows = 5;

int search\_element = 40;

int ans[] = Saddleback(nums, rows - 1, 0, search\_element);

System.out.println("Position of "+search\_element+" in the matrix is ("+ans[0] + "," + ans[1]+")");

}

private static int[] Saddleback(int nums[][], int row, int col, int search\_element) {

int element\_pos[] = {-1, -1};

if (row < 0 || col >= nums[row].length) {

return element\_pos;

}

if (nums[row][col] == search\_element) {

element\_pos[0] = row;

element\_pos[1] = col;

return element\_pos;

}

else if (nums[row][col] > search\_element) {

return Saddleback(nums, row - 1, col, search\_element);

}

return Saddleback(nums, row, col + 1, search\_element);

}

}

