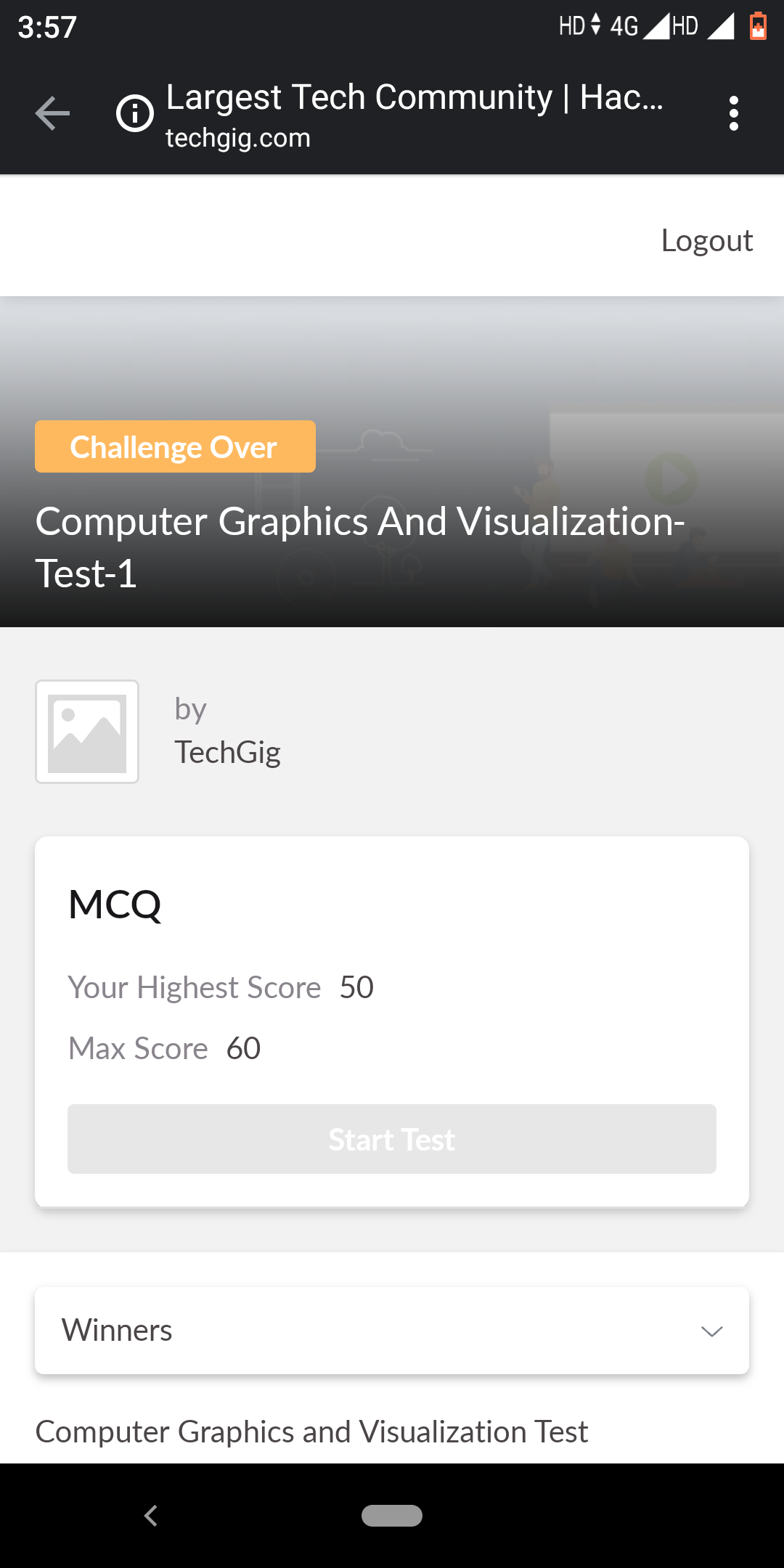
**DAILY ONLINE ACTIVITIES SUMMARY**

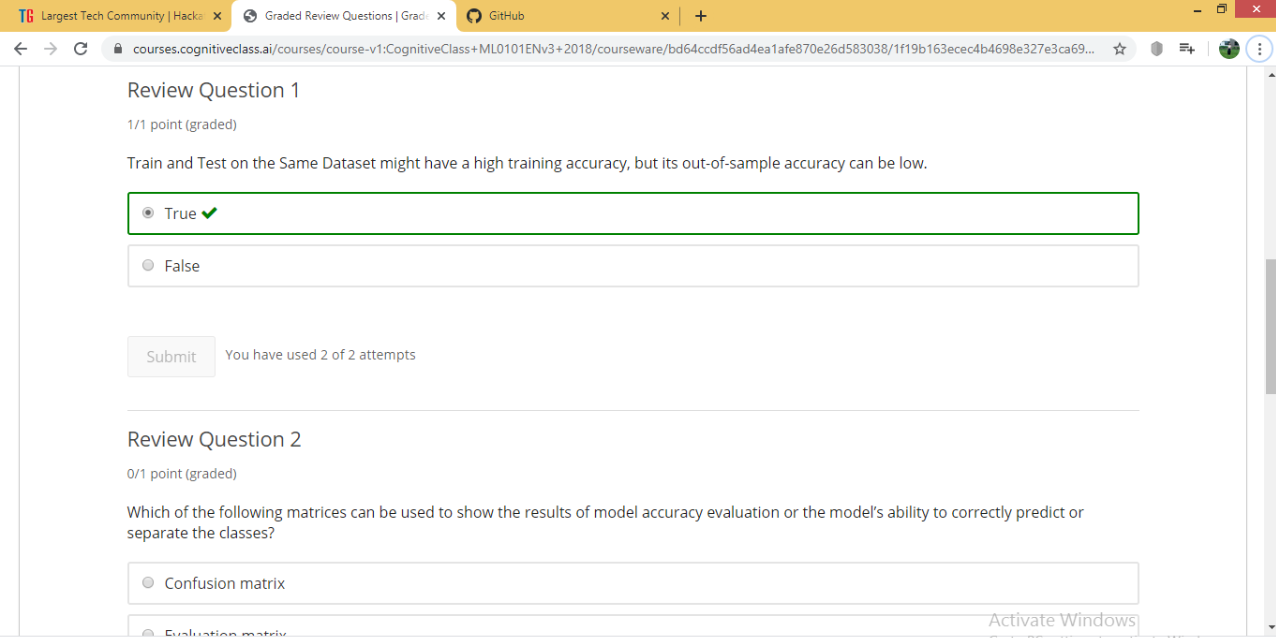
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19-05-2020** | | | | | **Name:** | **Shilpa S.U** | |
| **Sem & Sec** | **6th sem & B sec** | | | | | **USN:** | **4AL17CS090** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **Computer Graphics and Visualization** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **50** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Machine Learning with python** | | | | | | | |
| **Certificate Provider** | | | **Congitive class** | | **Duration** | | | **10 hours** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:** 1. We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome For example we take "S": S will be the shortest palindrome string. If we take "xyz": zyxyz will be the shortest palindrome string So we need to add some characters to the given string or character and 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. First take a Stack. Traverse through each node of the linked list and push each node value to Stack. Once the traversal & copying is done, iterate through linked list from head node again. In each iteration, pop one stack element and compare with node value in respective iteration. It is expected to match stack popped value with node value. In case of all matches, its a palindrome. Any one element mismatch makes it not a palindrome.  3.A user will input two strings, and we find if one of the strings is a sub sequence of the other. Program prints “yes” if either the first string is a sub sequence of the second string or the second string is a sub sequence of the first string. Assume that, the length of the first string is smaller than or equal to the length of the second string. | | | | | | | | |
| **Status: DONE** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **YES** | | | |
| **If yes Repository name** | | | | | **DAILY STATUS,QUARENTINE CODING,PYTHON CERTIFICATION COURSE** | | | |
| **Uploaded the report in slack** | | | | | **YES** | | | |

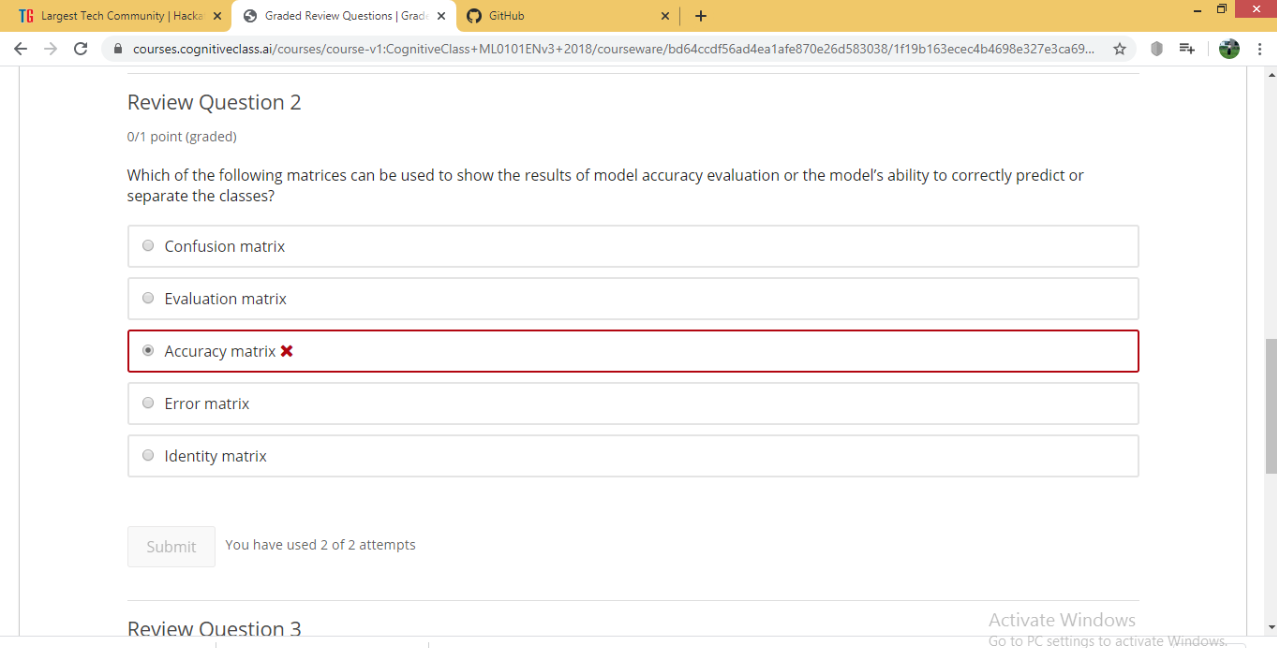
Online Test Details: (Attach the snapshot and briefly write the report for the same)

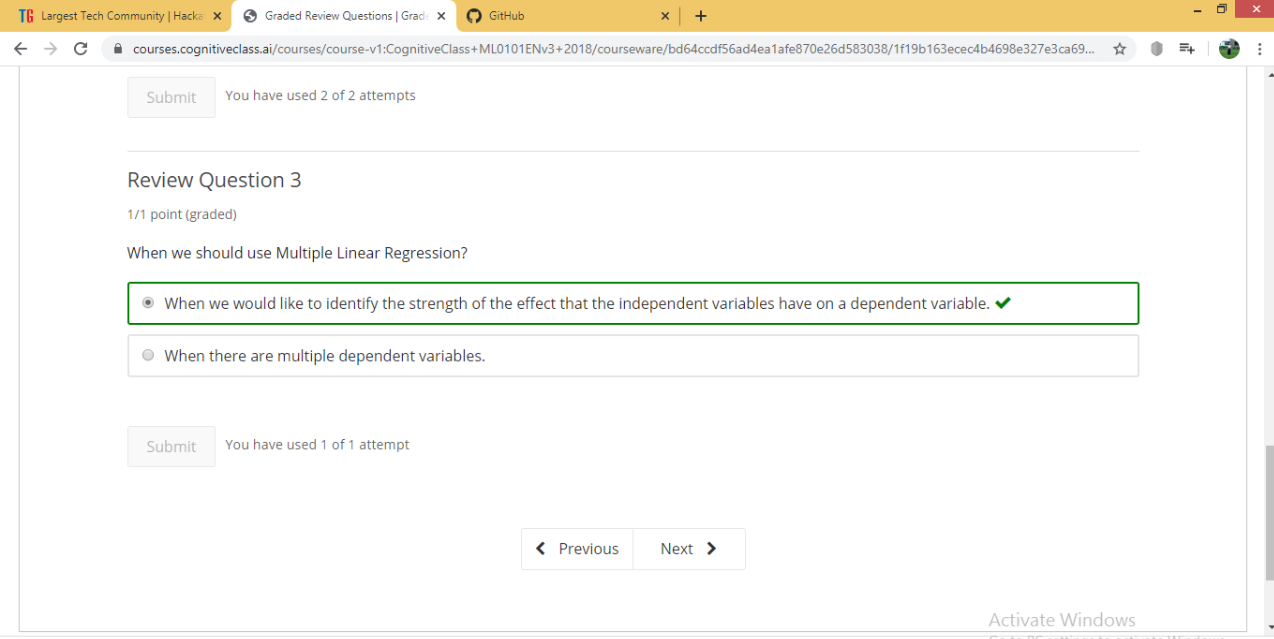


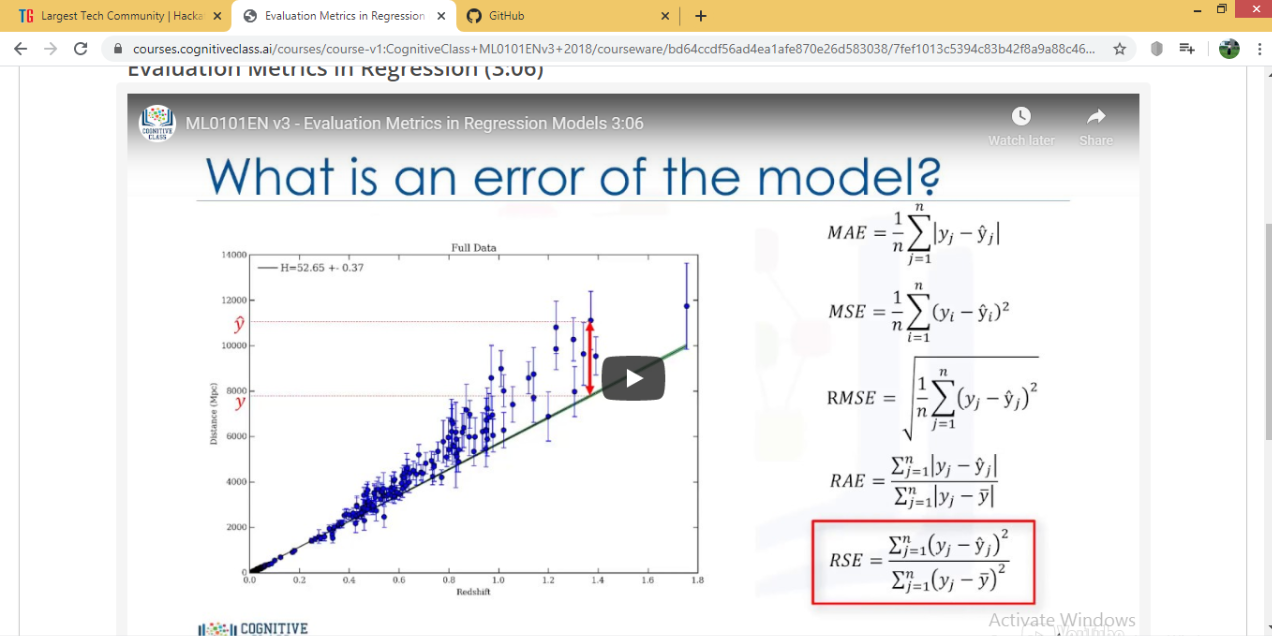
CGV test was held today i.e 19-05-2020. The maximum marks was for 60 marks. Out of which I got 54 marks

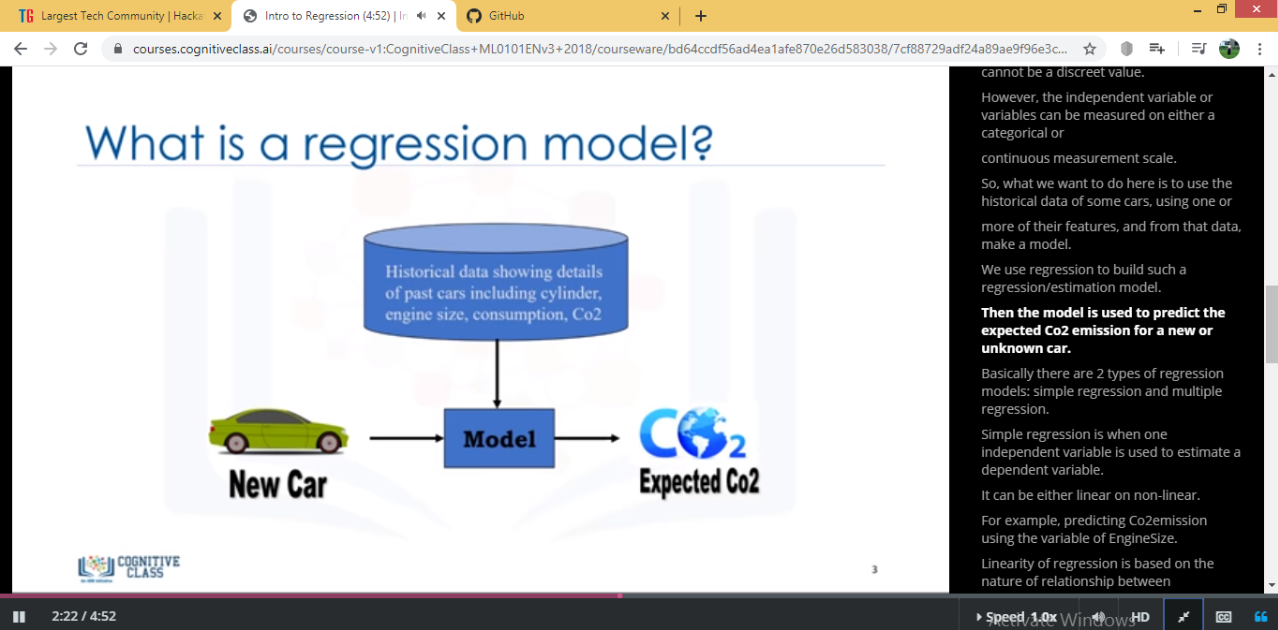
Certification Course Details: (Attach the snapshot and briefly write the report for the same)

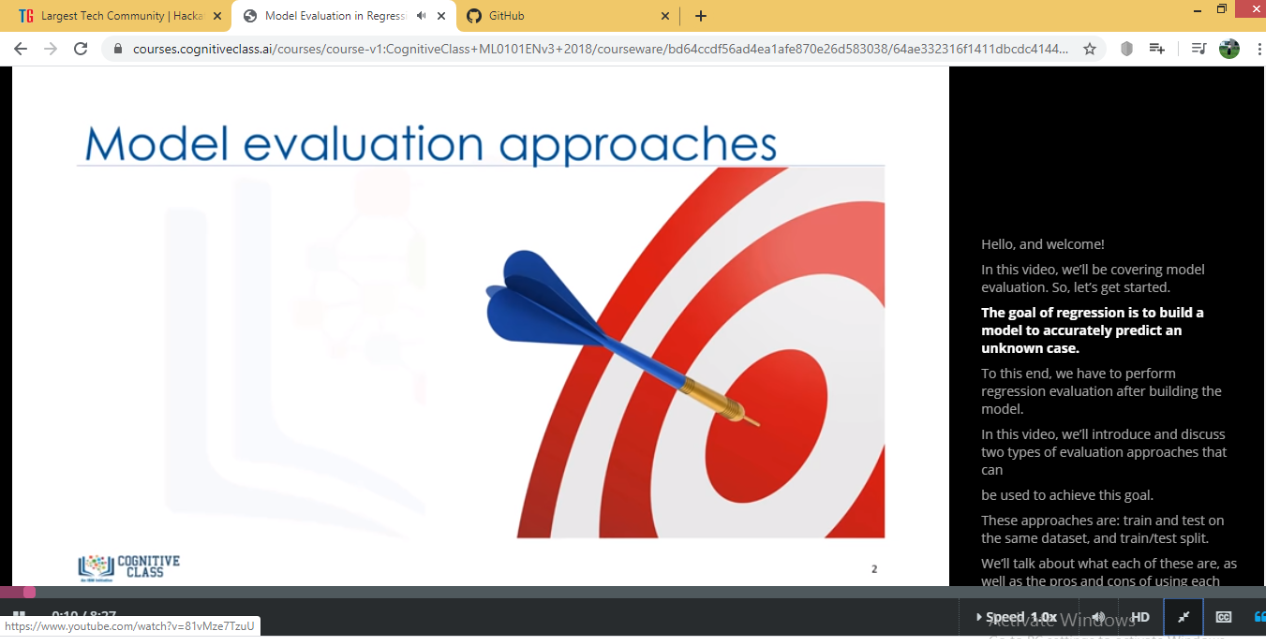


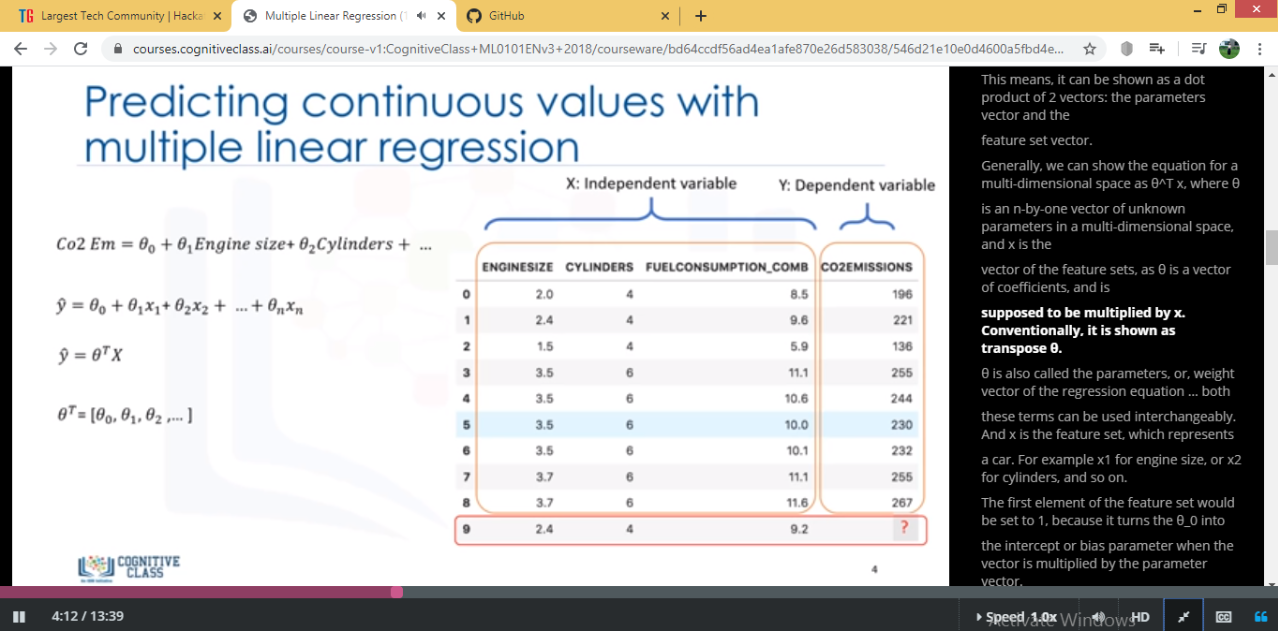


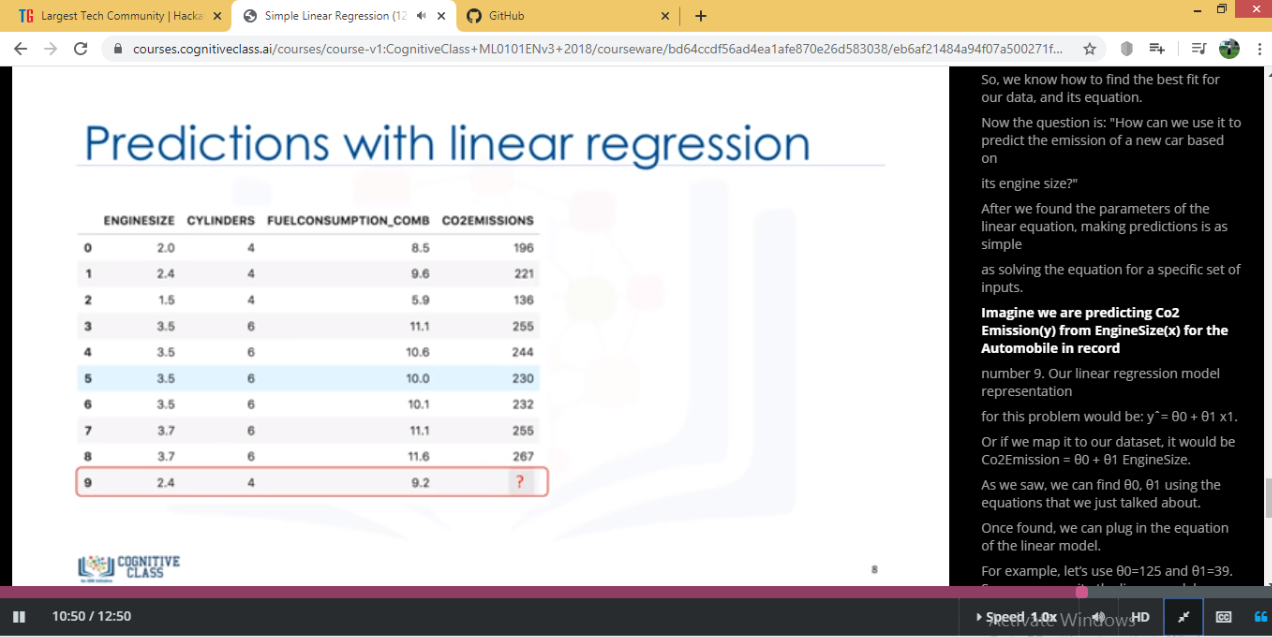






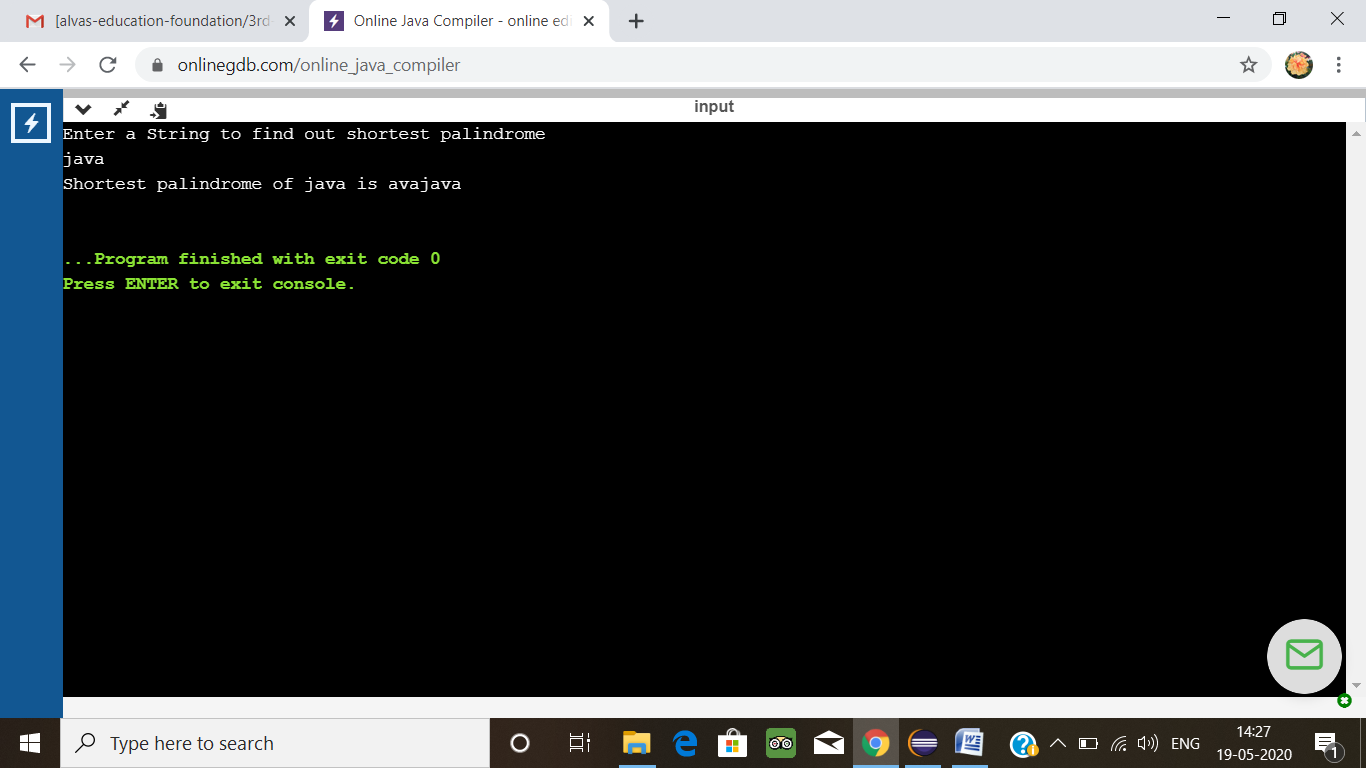


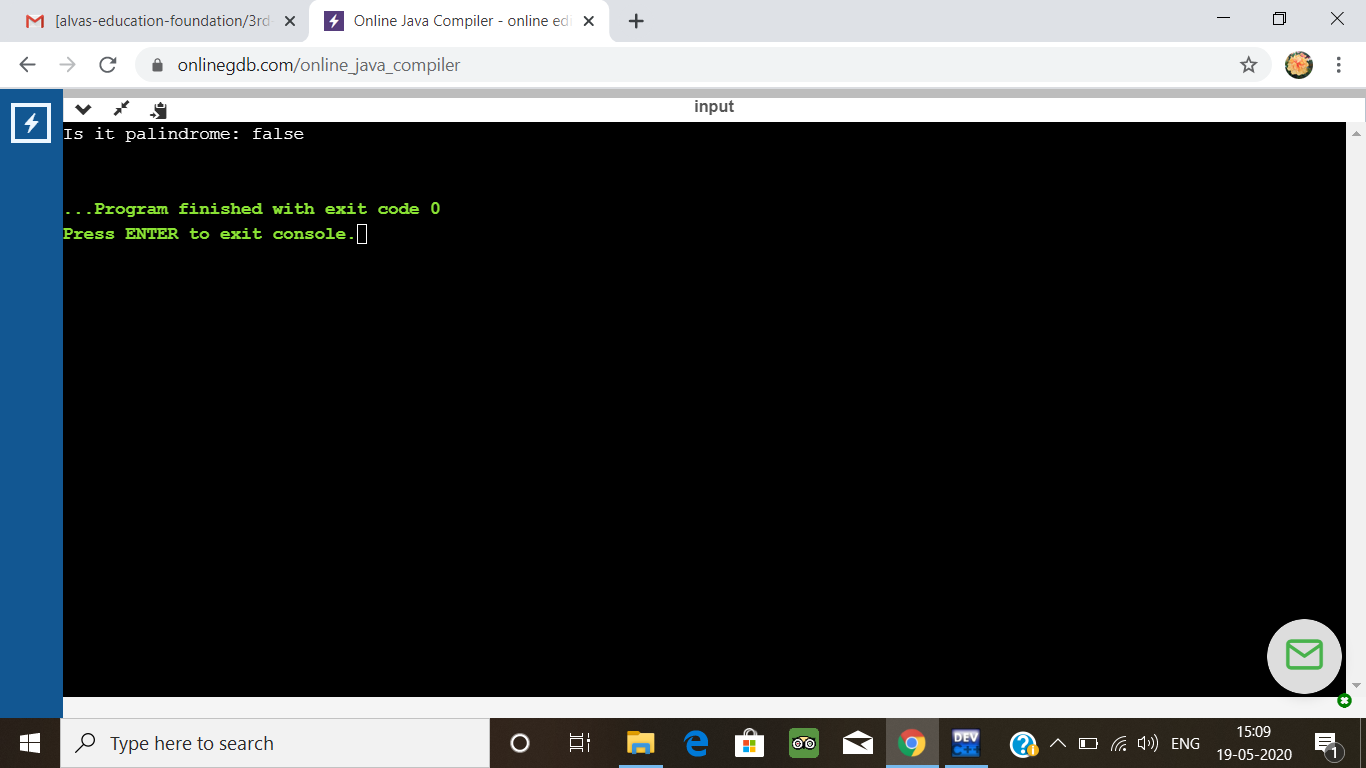




DAY 2 (19-05-2020)-Intro to Regression ,Simple Linear Regression ,Multiple Linear Regression ,Model Evaluation in Regression Models ,Evaluation Metrics in Regression ,Non-Linear Regression ,Graded Review Questions ARE COMPLETED

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

****

****

3.A user will input two strings, and we find if one of the strings is a sub sequence of the other. Program prints “yes” if either the first string is a sub sequence of the second string or the second string is a sub sequence of the first string.  
Assume that, the length of the first string is smaller than or equal to the length of the second string.

**An expected output of the program:**

Input the first string  
tree  
Input the second string  
Computer science is awesome  
YES

#include <stdio.h>  
#include <string.h>

int check\_subsequence (char [], char[]);

int main () {  
  int flag;  
  char s1[1000], s2[1000];

  printf("Input first string**\n**");  
  gets(s1);

  printf("Input second string**\n**");  
  gets(s2);

  if (strlen(s1) < strlen(s2))  
    flag = check\_subsequence(s1, s2);  
  else  
    flag = check\_subsequence(s2, s1);

  if (flag)  
    printf("YES**\n**");  
  else  
    printf("NO**\n**");

  return 0;  
}

int check\_subsequence (char a[], char b[]) {  
  int c, d;

  c = d = 0;

  while (a[c] != '**\0**') {  
    while ((a[c] != b[d]) && b[d] != '**\0**') {  
      d++;  
    }  
    if (b[d] == '**\0**')  
      **break**;  
    d++;  
    c++;  
  }  
  if (a[c] == '**\0**')  
    return 1;  
  else  
    return 0;  
}

**output**

