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

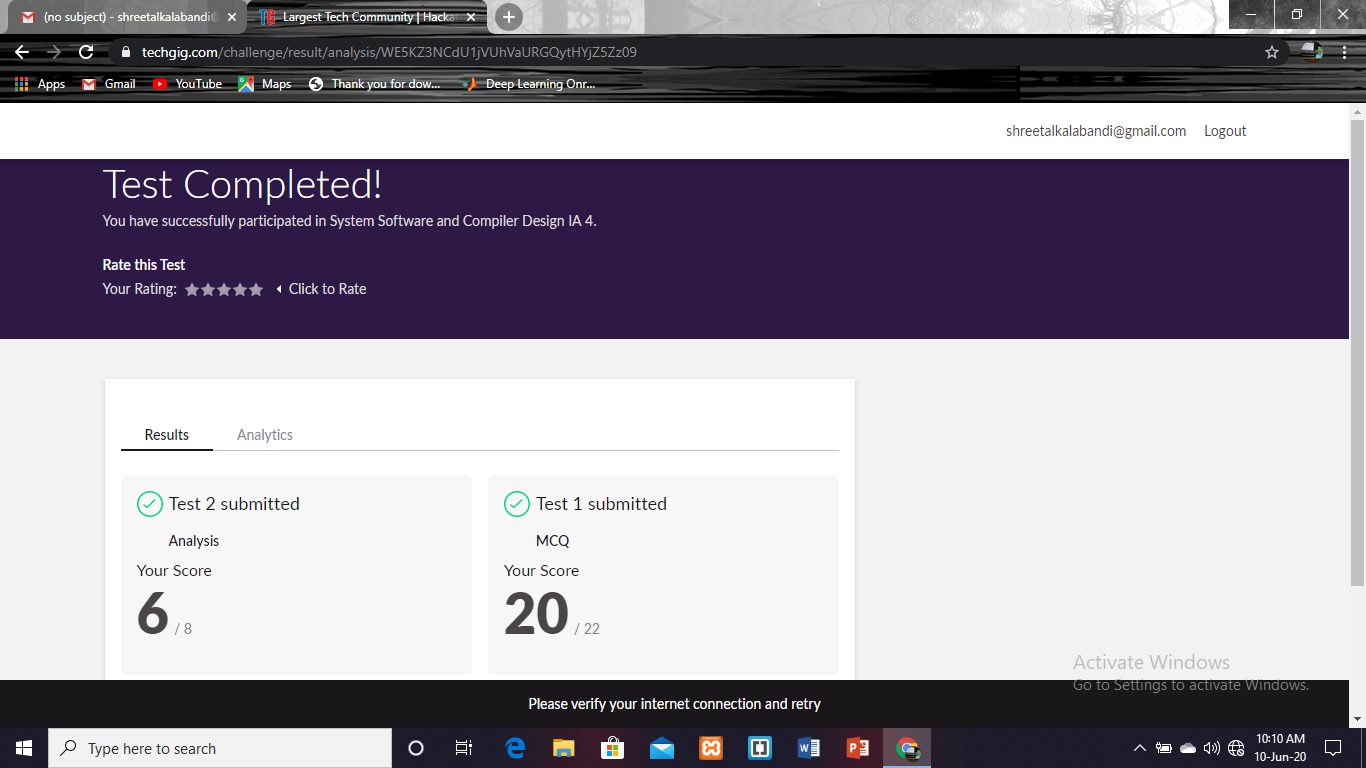
**Student Name: Shreetal kalabandi**

**Class and Sec: VI B**

**USN: 4AL17CS091**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Online Test Details** | | | | |
| **Subject** | **System Software and Compiler Design** | | | |
| **Semester** | **VI - B** | | **Duration** | **45 Minutes** |
| **87%** | | **26/30** | | |

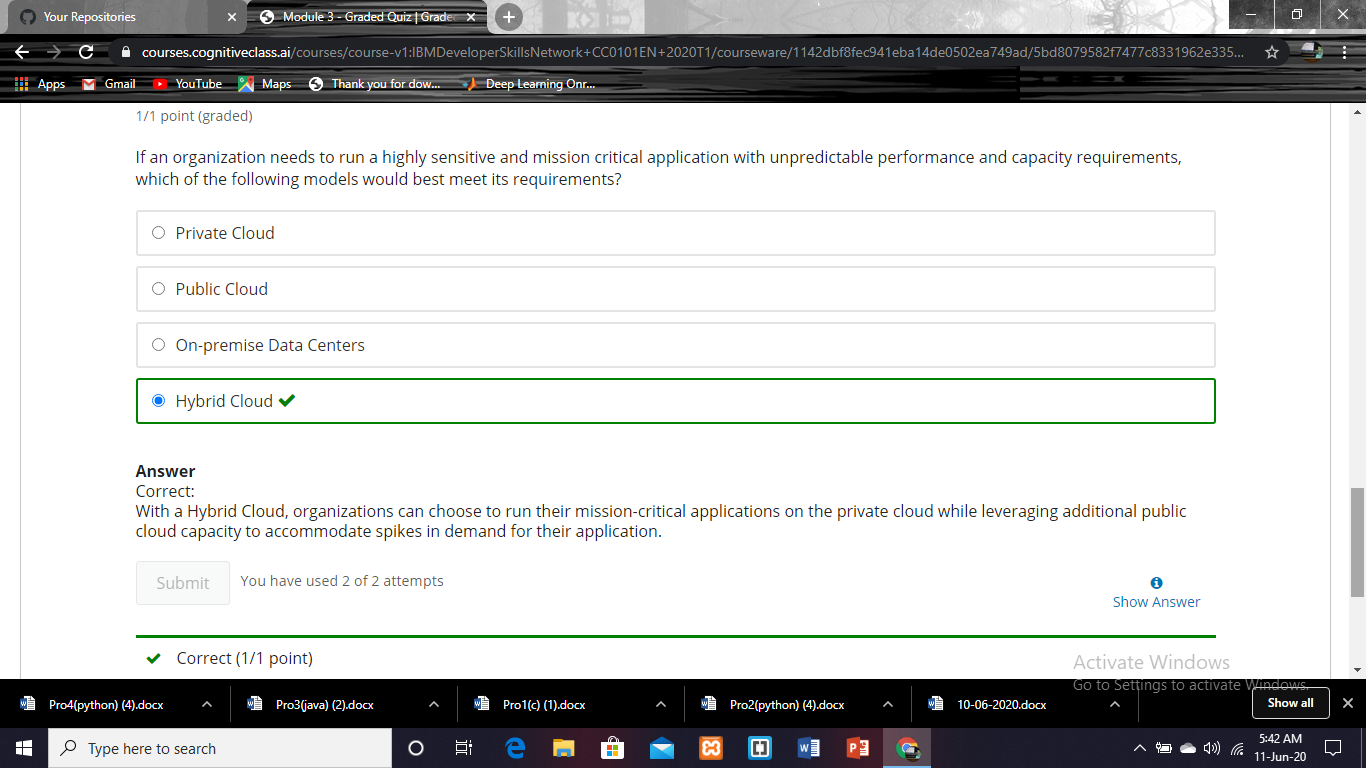
**Encl: snapshot of the test result**



|  |  |  |  |
| --- | --- | --- | --- |
| **Certification Course Details** | | | |
| **Course** | **Introduction to cloud** | | |
| **Certificate Provider** | **Cognitioclass.ai** | **Duration** | **6hrs** |

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

**Progress on 10-06-2020**

****

|  |  |
| --- | --- |
| **Coding Challenges** | |
| **Problem Statement: Pro1(c), Pro2(python), Pro3(java), Pro4(python).** | |
| **Status: Completed** | |
| **Uploaded the report both in GitHub & Slack** | **Yes** |

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

**1. Write a C Program to print the sum of boundary elements of a matrix.**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a[100][100],m,n,i,j;

printf("Enter The Size Of The Matrix:\n");

scanf("%d%d",&m,&n);

printf("Enter The Elements Into Matrix:\n");

for(i=0;i<m;i++)

{

for(j=0;j<n;j++)

{

scanf("%d",&a[i][j]);

}

}

int f,g;

printf("The Input Matrix Is:\n");

for(f=0;f<m;f++)

{

for(g=0;g<n;g++)

{

printf("%d\t",a[f][g]);

}

printf("\n");

}

printf("The Boundary Elements Are:\n");

int b,c,s=0;

for(b=0;b<m;b++)

{

for(c=0;c<n;c++)

{

if(b==0 || b==m-1)

{

s=s+a[b][c];

printf("%d\t",a[b][c]);

}

else if(c==0 || c==n-1)

{

s=s+a[b][c];

printf("%d\t",a[b][c]);

}

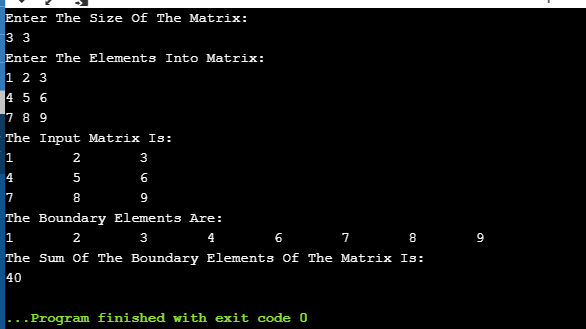
}

}

printf("\nThe Sum Of The Boundary Elements Of The Matrix Is:\n%d",s);

}

**Output:**

****

**2.** **Python Program to find the length of the list using Recursion.**

def length(a):

if len(a)==0:

return 0

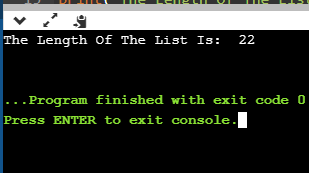
else:

return 1+length(a[1:])

a = [2, 4, 8, 4, 6, 3, 5, 7, 2, 3, 8, 1, 3, 6, 9, 4, 3, 5, 8, 4, 1, 2]

print("The Length Of The List Is: ", length(a))

**Output:**

****

**3. Write a Java program to find the maximum and minimum value node from a circular linked list.**

package pblm;

import java.util.\*;

public class MinMax {

public class Node {

int data;

Node next;

public Node(int data)

{

this.data = data;

}

}

public Node head = null;

public Node tail = null;

public void add(int data)

{

Node newNode = new Node(data);

if(head == null)

{

head = newNode;

tail = newNode;

newNode.next = head;

}

else

{

tail.next = newNode;

tail = newNode;

tail.next = head;

}

}

public void minNode()

{

Node current = head;

int min = head.data;

if(head == null)

{

System.*out*.println("List Is Empty");

}

else {

do{

if(min > current.data)

{

min = current.data;

}

current= current.next;

}while(current != head);

System.*out*.println("Minimum Value Node In The List: "+ min);

}

}

public void maxNode()

{

Node current = head;

int max = head.data;

if(head == null)

{

System.*out*.println("List Is Empty");

}

else {

do{

if(max < current.data)

{

max = current.data;

}

current= current.next;

}while(current != head);

System.*out*.println("Maximum Value Node In The List: "+ max);

}

}

public static void main(String[] args) {

Scanner s=new Scanner(System.*in*);

MinMax cl = new MinMax();

System.*out*.println("Enter The Number Of Elements: ");

int n=s.nextInt();

System.*out*.println("Enter The Elements Into Linked List");

for(int i=0;i<n;i++)

{

cl.add(s.nextInt());

}

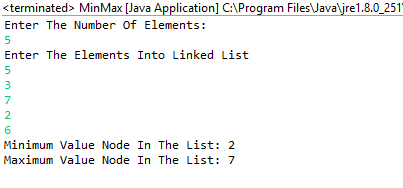
cl.minNode();

cl.maxNode();

}

}

**Output:**



**4. Python Program to check whether a given number is a fibonacci number or not .**

import math

def persq(a):

if int(math.sqrt(a))\*\*2 == a:

return True

else:

return False

n = int(input("Enter The Number: "))

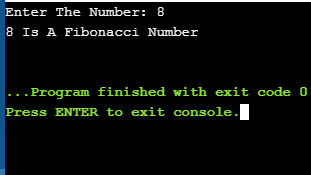
if persq(5\*(n\*\*2)+4) or persq(5\*(n\*\*2)-4):

print(n, "Is A Fibonacci Number")

else:

print(n, "Is Not A Fibonacci Number")

**Output:**

****