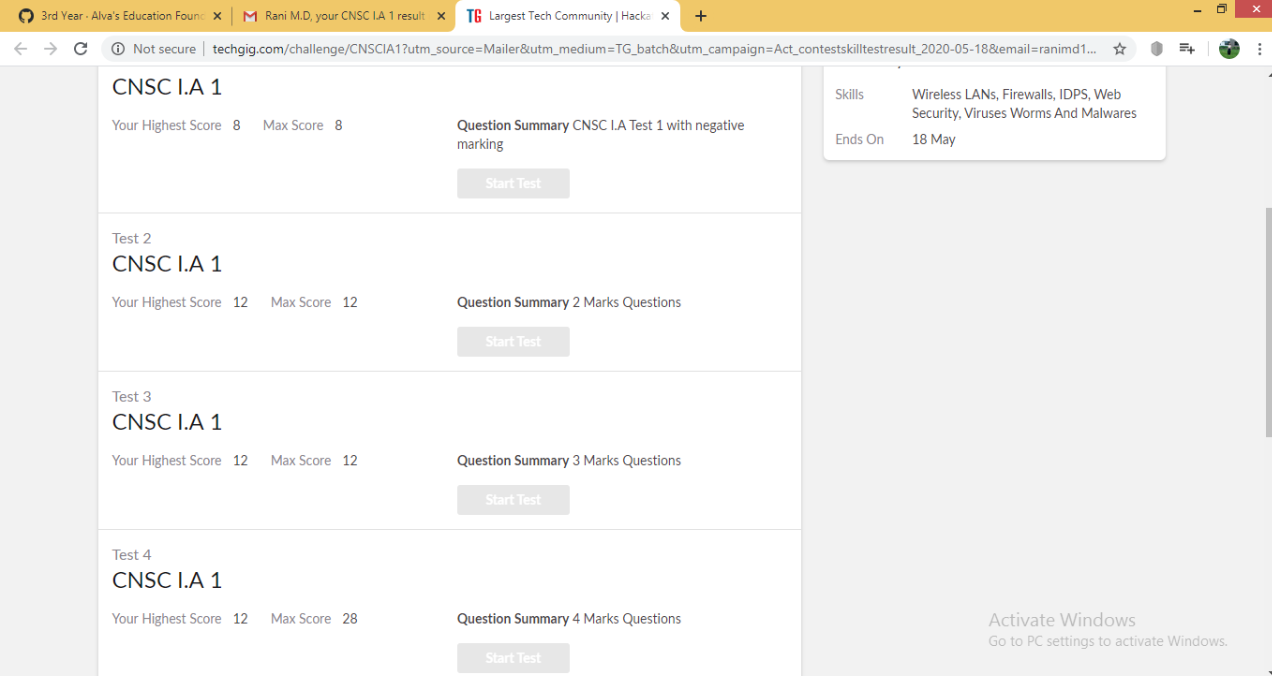
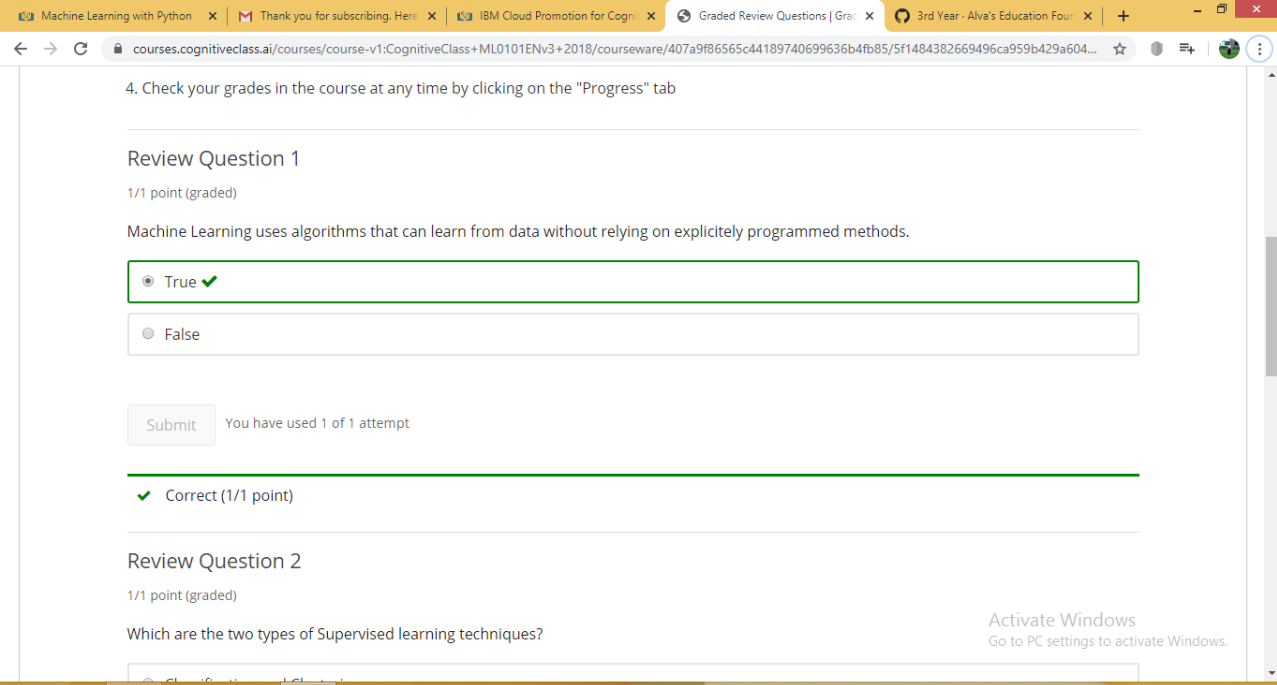
**DAILY ONLINE ACTIVITIES SUMMARY**

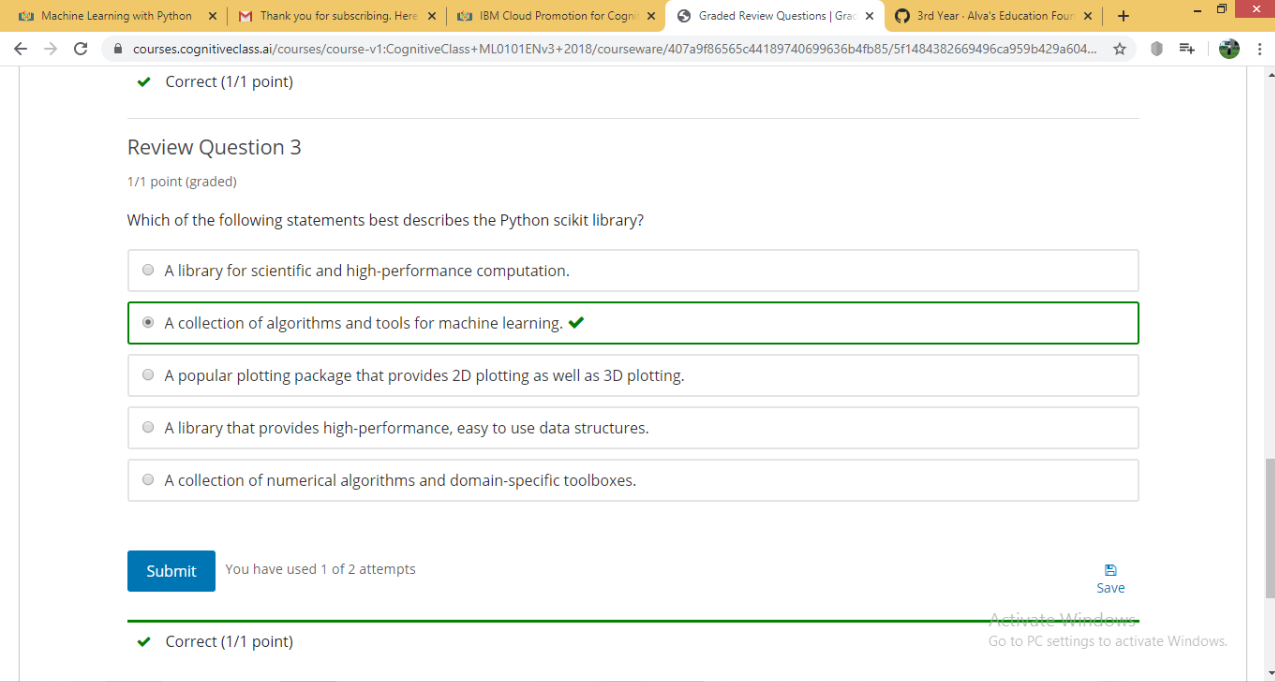
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18 May 2020** | | | | | **Name:** | **Rani M.D** | |
| **Sem & Sec** | **6th sem & B sec** | | | | | **USN:** | **4AL17CS075** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **Cryptography Network Security & Cyber Laws** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **44** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Machine Learning with python** | | | | | | | |
| **Certificate Provider** | | | **Congnitive Class** | | **Duration** | | | **10 hours** |
| **Coding Challenges** | | | | | | | | |
| 1. **Problem Statement:** Using methods charAt() & length() of String class, write a program to print the frequency of each character in a string.   “Hello friend”  Output should be -: 1 d: 1 e: **2** f: 1 (continued for all character in the string)   1. Write down a java program to print even and odd numbers series respectively from two threads: t1 and t2 synchronizing on a shared object   Let t1 print message “ping — >” and t2 print message “,—-pong”.  Take as command line arguments, the following inputs to the program: Sleep Interval for thread t1 Write down a java program to print even and odd numbers series respectively from two threads: t1 and t2 synchronizing on a shared object  Let t1 print message “ping — >” and t2 print message “,—-pong”.  Take as command line arguments, the following inputs to the program:  Sleep Interval for thread t1  Sleep Interval for thread t2  Message per cycle  No of cycles | | | | | | | | |
| **Status: DONE** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **YES** | | | |
| **If yes Repository name** | | | | | **Quarentine coding** | | | |
| **Uploaded the report in slack** | | | | |  | | | |

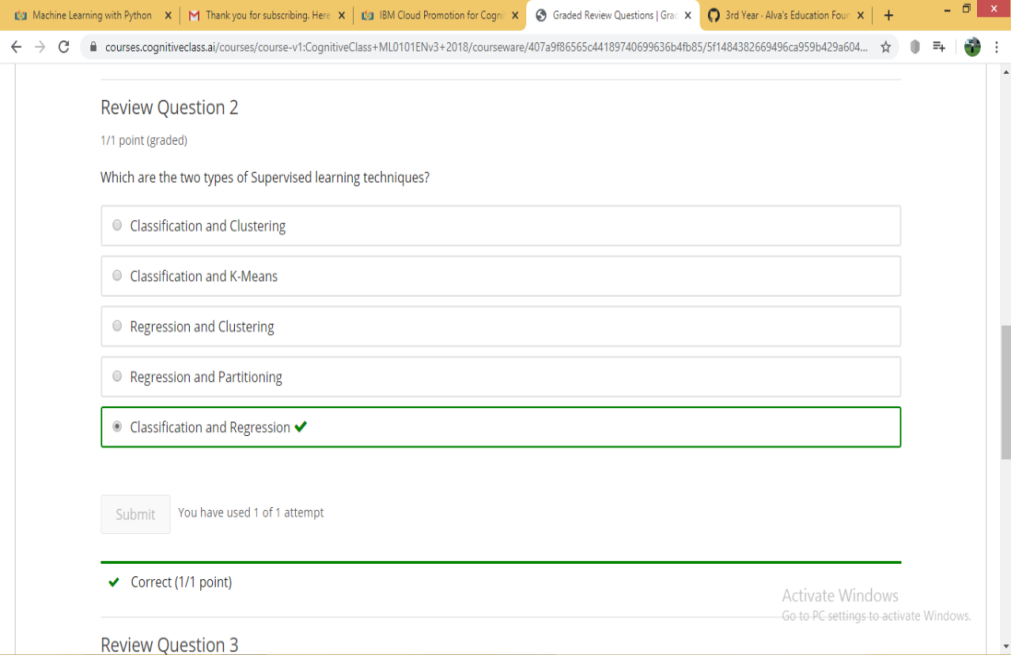
Online Test Details: (Attach the snapshot and briefly write the report for the same) 

CNSC test was held today i.e 18 May 2020. There were four rounds where each round carried 8,12,12,28 marks respectively. Out of 60 marks I scored 44

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







DAY 1 (18-05-2020)- INTRODUCTION ,GENERAL OBJECTIVES ,MODULE 1 MACHINE LEARNING AND REVIEW QUESTIONS ARE COMPLETED

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

Program 1

import java.util.Scanner;

public class Main

{

public static void main(String args[])

{

int i;

String str;

int counter[] = new int[256];

Scanner in = new Scanner(System.in);

System.out.print("Enter a String : ");

str=in.nextLine();

for (i = 0; i < str.length(); i++) {

counter[(int) str.charAt(i)]++;

}

for (i = 0; i < 256; i++) {

if (counter[i] != 0) {

System.out.println( (char) i + ":" +counter[i]);

}

}

}

}

Program 2

class OddThread extends Thread

{

int limit;

sharedPrinter printer;

public OddThread(int limit, sharedPrinter printer)

{

this.limit = limit;

this.printer = printer;

}

@Override

public void run()

{

int oddNumber = 1;

while (oddNumber <= limit)

{

printer.printOdd(oddNumber);

oddNumber = oddNumber + 2;

}

}

}

class EvenThread extends Thread

{

int limit;

sharedPrinter printer;

public EvenThread(int limit, sharedPrinter printer)

{

this.limit = limit;

this.printer = printer;

}

@Override

public void run()

{

int evenNumber = 2;

while (evenNumber <= limit)

{

printer.printEven(evenNumber);

evenNumber = evenNumber + 2;

}

}

}

class sharedPrinter

{

boolean isOddPrinted = false;

synchronized void printOdd(int number)

{

while (isOddPrinted)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName()+" "+number);

isOddPrinted = true;

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

notify();

}

synchronized void printEven(int number)

{

while (! isOddPrinted)

{

try

{

wait();

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

System.out.println(Thread.currentThread().getName()+" "+number);

isOddPrinted = false;

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

notify();

}

}

public class Main

{

public static void main(String[] args)

{

sharedPrinter printer = new sharedPrinter();

OddThread oddThread = new OddThread(20, printer);

oddThread.setName("—-pong");

EvenThread evenThread = new EvenThread(20, printer);

evenThread.setName("ping — >");

oddThread.start();

evenThread.start();

}

}