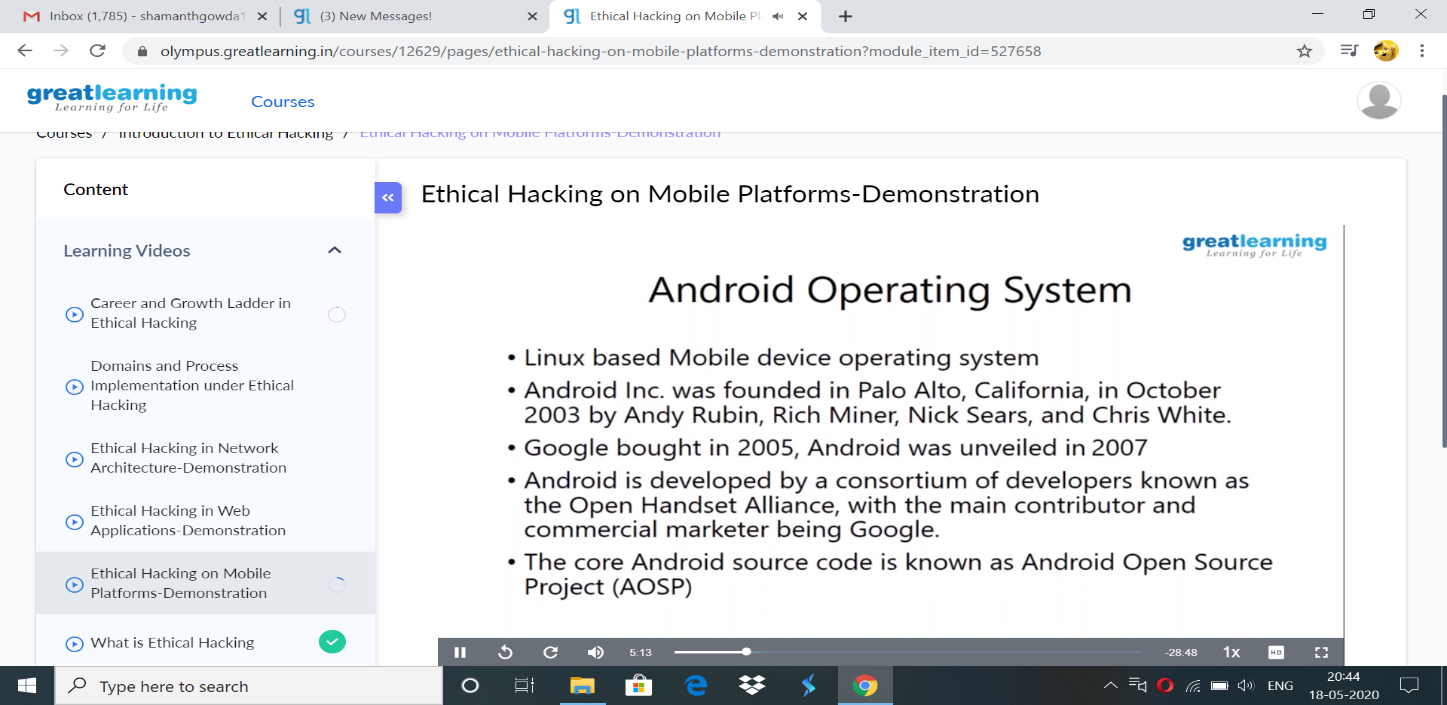
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18-05-2020** | | | | | **Name:** | **Tanoj .M** | |
| **Sem & Sec** | **VI A** | | | | | **USN:** | **4AL16CS113** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **CNSC IA Test** | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | **46** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **ETHICAL HACKING** | | | | | | | |
| **Certificate Provider** | | | **Great learning** | | **Duration** | | | **6 days** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  **1.** Using methods charAt() & length() of String class, write a program to print the frequency of each character in a string.  **2.** 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”. . | | | | | | | | |
| **Status:Completed** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **Yes** | | | |
| **If yes Repository name** | | | | | **https://github.com/Tanoj8296/DAILY-STATUS** | | | |
| **Uploaded the report in slack** | | | | | **Yes** | | | |

**Online Certification Details**

Lesson-1

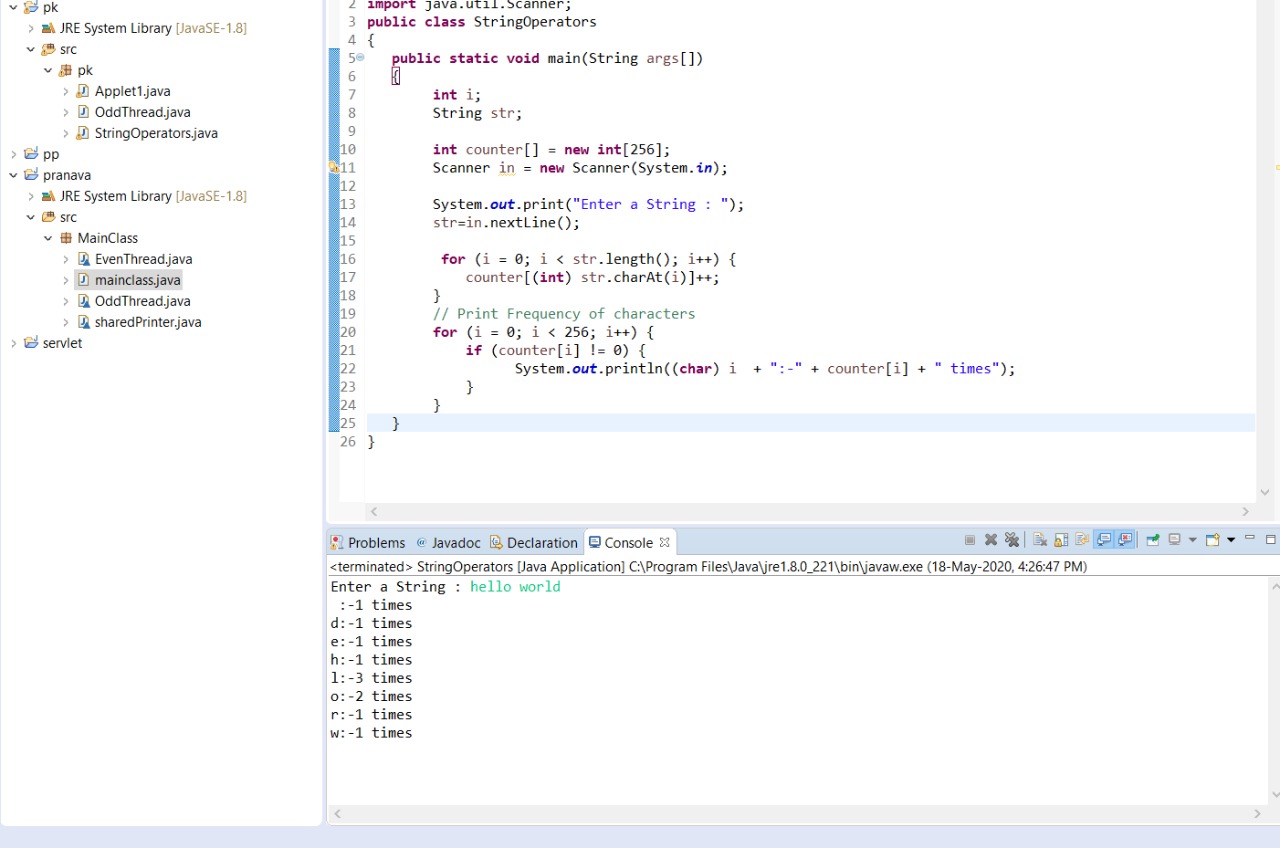
* What is Ethical Hacking?



* Itroduction to Ethical hacking
* Computer Security Threats
* Goals Skills and tools utilized
* Process Flow for an Ethical hack
* Demonstration
* Ethical Hacking across Domanis

**Coding Challenge Details**

1. Using methods charAt() & length() of String class, write a program to print the frequency of each character in a 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”.

package MainClass;

import MainClass.sharedPrinter;

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; //First odd number is 1

while (oddNumber <= limit)

{

printer.printOdd(oddNumber);

oddNumber = oddNumber + 2; //Incrementing to next odd number

}

}

}

package MainClass;

import MainClass.sharedPrinter;

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; //First even number is 2

while (evenNumber <= limit)

{

printer.printEven(evenNumber); //Calling printEven() method of SharedPrinter class

evenNumber = evenNumber + 2; //Incrementing to next even number

}

}

package MainClass;

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();

}

}

package MainClass;

public class mainclass

{

public static void main(String[] args)

{

sharedPrinter printer = new sharedPrinter();

OddThread oddThread = new OddThread(20, printer);

oddThread.setName("ping-->");

EvenThread evenThread = new EvenThread(20, printer);

evenThread.setName("pong-->");

oddThread.start();

evenThread.start();

}

}

OUTPUT:

