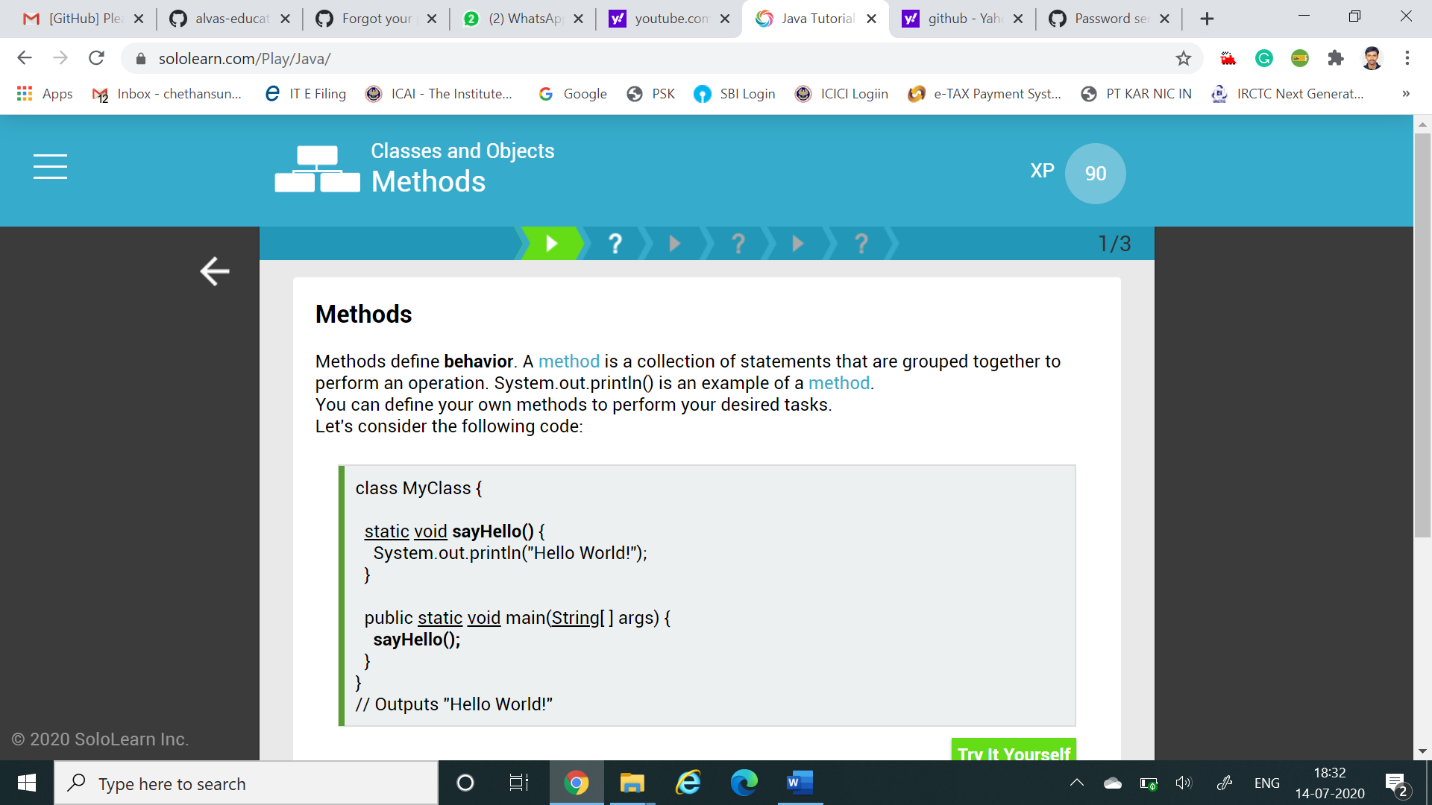
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **8/07/2020** | | | **Name:** | **Sannidhi P** | |
| **Sem & Sec** | **8th- B** | | | **USN:** | **4AL16CS084** | |
| **Certification Course Summary** | | | | | | |
| **Course** | **JAVASCRIPT Tutorial** | | | | | |
| **platform** | | **SOLOLEARN** | **Duration** | | | **2 hours** |
| **Coding Challenges** | | | | | | |
| **Problem Statement:**  **program with multithreaded program using Runnable interface**  **?** | | | | | | |
| **Status: Executed** | | | | | | |
| **Uploaded the report in Github** | | | **Yes** | | | |
| **If yes Repository name** | | | **Sannidhi1** | | | |
| **Uploaded the report in slack** | | | **Yes** | | | |

Certification:



Coding Challenges Details:

class ThreadX implements Runnable{

public void run( ) {

for(int i = 1; i <= 5; i++) {

System.out.println("Thread X with i = "+ i);

}

System.out.println("Exiting Thread X ...");

}

}

class ThreadY implements Runnable{

public void run( ) {

for(int j = 1; j <= 5; j++) {

System.out.println("Thread Y with j = "+ j);

}

System.out.println("Exiting Thread Y ...");

}

}

class ThreadZ implements Runnable{

public void run( ) {

for(int k = 1; k <= 5; k++) {

System.out.println("Thread Z with k = "+ k);

}

System.out.println("Exiting Thread Z ...");

}

}

class MultiThreadRunnable{

public static void main(String args[]) {

ThreadX x = new ThreadZ(); Thread t1 = new Thread(x);

ThreadY y = new ThreadY(); Thread t2 = new Thread(y);

ThreadZ z = new ThreadZ(); Thread t3 = new Thread(z);

t1.start();

t2.start();

t3.start();

System.out.println("... Multithreading is over ");

}

}