LAB_ANP_C6339_CLASSES_ StudenID:AF0339439

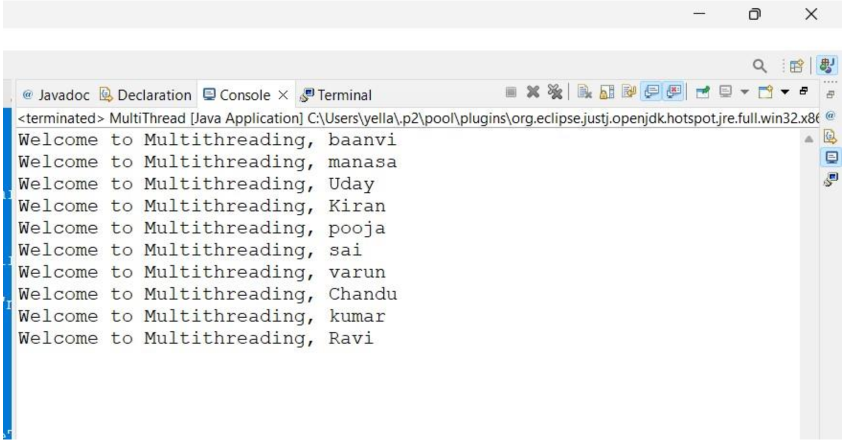
YELLA UDAY KUMAR

Assignment-1 create a multithreading program to display the given name with welcome message display greeting for available users. store 5 names using array of string, pass the string to the methods to diplay greeting message. create 2 threads to perform the above task.

Program:

```
package Thread;


class WelcomeThread extends Thread {

   String[] names;

    public  WelcomeThread(String[] names) {
        this.names = names;
    }

    public void run()
    {
        for (String name : names)
        {
            displayWelcomeMessage(name);
        }
    }
   public void displayWelcomeMessage(String name) {
        System.out.println("Welcome to Multithreading, " + name );
    }
}

public class MultiThread
{
    public static void main(String[] args)
    {

   String[] names1 = {"Uday", "Kiran", "Chandu", "kumar", "Ravi"};

  String[] names2 = {"baanvi", "manasa", "pooja", "sai", "varun"};




        WelcomeThread t1 = new WelcomeThread(names1);
        WelcomeThread t2 = new WelcomeThread(names2);
        t1.start();
        t2.start();


        try {
            t1.join();
            t2.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

    }
}
```

Output:



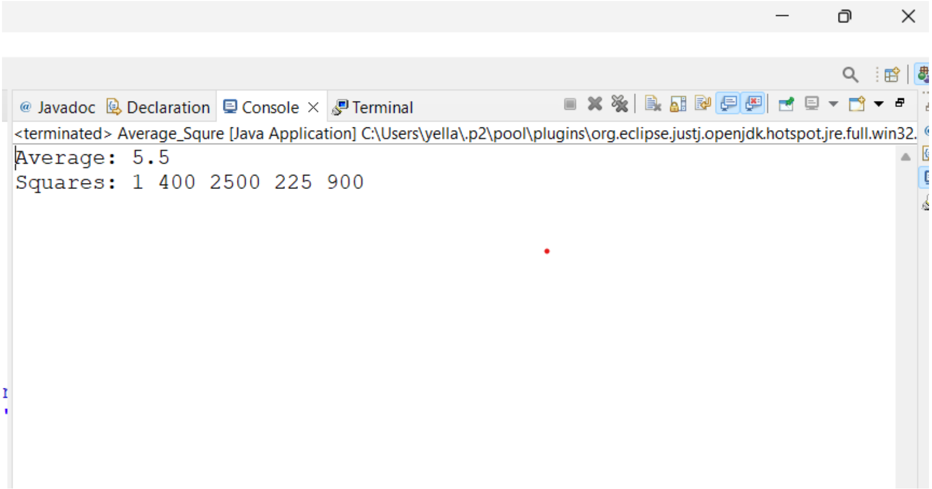
```
<terminated> MultiThread [Java Application] C:\Users\yella\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
Welcome to Multithreading, baanvi
Welcome to Multithreading, manasa
Welcome to Multithreading, Uday
Welcome to Multithreading, Kiran
Welcome to Multithreading, pooja
Welcome to Multithreading, sai
Welcome to Multithreading, varun
Welcome to Multithreading, Chandu
Welcome to Multithreading, kumar
Welcome to Multithreading, Ravi
```

Assignment 2 Create two thread.one thread is finding the average of the first 10 numbers andanother thread is printing the square of the number stored in array arr={1,20,50,15,30}and make sure both threads can execute one by one.

Program:

```
package Thread;
class AverageThread extends Thread {
    private int[] numbers;

    public AverageThread(int[] numbers) {
        this.numbers = numbers;
    }

    @Override
    public void run() {
        synchronized (this){
               int sum = 0;
            for (int i = 0; i <=10; i++) {
              sum = sum+i;
            }
            double average = sum / 10.0;
            System.out.println("Average: " + average);
        }
    }
}

class SquareThread extends Thread {
    private int[] numbers;

    public SquareThread(int[] numbers) {
        this.numbers = numbers;
    }

    @Override
    public void run() {
        synchronized (this) {
            System.out.print("Squares: ");
            for (int i = 0; i < 5; i++) {
                int square = numbers[i] * numbers[i];
                System.out.print(square + " ");
            }
            System.out.println();
        }
    }
}

public class Average_Squre {
    public static void main(String[] args) {
        int[] numbers = {1, 20, 50, 15, 30};

        AverageThread averageThread = new AverageThread(numbers);
        SquareThread squareThread = new SquareThread(numbers);

        averageThread.start();
        try {
            averageThread.join(); // Wait for the averageThread to
finish before starting squareThread
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        squareThread.start();
    }
}
```

Output:



<terminated> Average_Squre [Java Application] C:\Users\yella\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.

```
Average: 5.5
Squares: 1 400 2500 225 900
```