

Lab17- 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 display greeting message.

create 2 threads to perform the above task.

```
package com.wrapperclass.examples;
class GreetingThread extends Thread {
    private String[] names;

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

    @Override
    public void run() {
        displayGreetings();
    }

    private void displayGreetings() {
        for (String name : names) {
            System.out.println("Welcome, " + name +
"!");
        }
    }
}

public class MultiThreadedGreeting {
    public static void main(String[] args) {
        // Array of names
        String[] names = {"Alice", "Bob", "Charlie",
"David", "Eve"};

        // Create two threads with the same set of
names
        GreetingThread thread1 = new
GreetingThread(names);
```

```

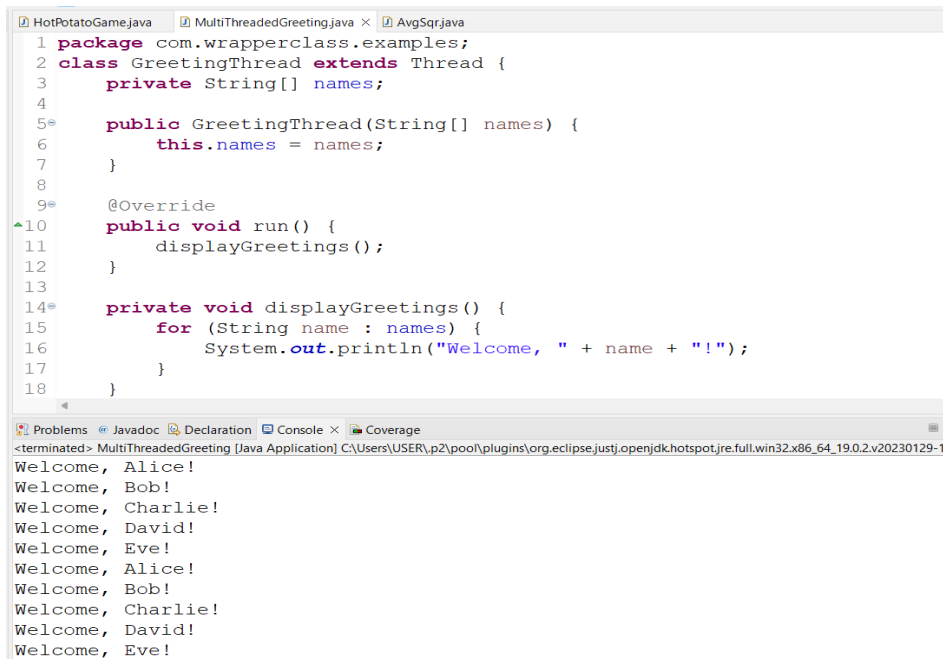
        GreetingThread thread2 = new
GreetingThread(names);

        // Start both threads
        thread1.start();
        thread2.start();
    }

}

```

Output:



The screenshot shows an IDE with two tabs: 'MultiThreadedGreeting.java' and 'AvgSqr.java'. The 'MultiThreadedGreeting.java' tab is active, displaying the following code:

```

1 package com.wrapperclass.examples;
2 class GreetingThread extends Thread {
3     private String[] names;
4
5     public GreetingThread(String[] names) {
6         this.names = names;
7     }
8
9     @Override
10    public void run() {
11        displayGreetings();
12    }
13
14    private void displayGreetings() {
15        for (String name : names) {
16            System.out.println("Welcome, " + name + "!");
17        }
18    }

```

Below the code editor, the 'Console' tab is active, showing the output of the program:

```

<terminated> MultiThreadedGreeting [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20230129-11
Welcome, Alice!
Welcome, Bob!
Welcome, Charlie!
Welcome, David!
Welcome, Eve!
Welcome, Alice!
Welcome, Bob!
Welcome, Charlie!
Welcome, David!
Welcome, Eve!

```

2) Create two thread.one thread is finding the average of the first 10 numbers and another 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.

```
package com.wrapperclass.examples;
```

```

class AverageThread extends Thread {
    @Override

    public void run() {

        double average = calculateAverage();
        System.out.println("Average: " + average);
    }

    private double calculateAverage() {
        double sum = 0;
        for (int i = 1; i <= 10; i++) {
            sum += i;
        }
        return sum / 10;
    }
}

class SquareThread extends Thread {
    private int[] arr = {1, 20, 50, 15, 30};

    @Override
    public void run() {
        System.out.print("Squares: ");
        for (int num : arr) {
            System.out.print(num * num + " ");
        }
        System.out.println();
    }
}

public class AvgSqr {
    public static void main(String[] args) {
        AverageThread averageThread = new
AverageThread();
        SquareThread squareThread = new SquareThread();

        averageThread.start();
        squareThread.start();
    }
}

```

Output:

```
HotPotatoGame.java MultiThreadedGreeting.java AvgSqr.java ×
1 package com.wrapperclass.examples;
2
3 class AverageThread extends Thread {
4     @Override
5
6     public void run() {
7
8         double average = calculateAverage();
9         System.out.println("Average: " + average);
10    }
11
12    private double calculateAverage() {
13        double sum = 0;
14        for (int i = 1; i <= 10; i++) {
15            sum += i;
16        }
17        return sum / 10;
18    }

```

Problems @ Javadoc Declaration Console × Coverage

<terminated> AvgSqr [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20230129-1

Squares: 1 400 2500 225 900

Average: 5.5