

Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering
Software Engineering Lab (CS29006)

Instructor: Prof. Sudip Misra

Assignment 4: Java Programming

Time: 2:00 pm - 4:55 pm

Date : 13/02/2019

Instructions for submission

- Give meaningful comments to explain the functionality of each class and function used in your program.
 - Make a zip file with and give the name of the zip file as **A4_ < YourRollNo >**.
 - The zip file should contain codes for the practice problems and the assignments.
 - Submit the zip file to the Moodle system.
 - Submit your solution latest by **4:55 pm** on **13/02/2019**.
-

1 Problems for Practice

1. Type the following code and check the output.

```
1 class RunnableThread implements Runnable {
2
3     @Override
4     public void run() {
5         System.out.println("RunnableThread - START " + Thread.currentThread().getName());
6         try {
7             Thread.sleep(1000);
8             // Get database connection, delete unused data from DB
9             doProcessing();
10        } catch (InterruptedException e) {
11            e.printStackTrace();
12        }
13        System.out.println("RunnableThread - END " + Thread.currentThread().getName());
14    }
15
16    private void doProcessing() throws InterruptedException {
17        Thread.sleep(5000);
18    }
19
20 }
21
22 class MyThread extends Thread {
23
24     public MyThread(String name) {
25         super(name);
26     }
27
28     @Override
29     public void run() {
30         System.out.println("MyThread - START " + Thread.currentThread().getName());
31         try {
32             Thread.sleep(1000);
33             // Get database connection, delete unused data from DB
34             doProcessing();
35        } catch (InterruptedException e) {
36            e.printStackTrace();
37        }
38        System.out.println("MyThread - END " + Thread.currentThread().getName());
39    }
40
41    private void doProcessing() throws InterruptedException {
42        Thread.sleep(5000);
43    }
44
45 }
46
```

```
47 public class ThreadExample {
48
49     public static void main(String[] args) {
50         Thread t1 = new Thread(new RunnableThread(), "t1");
51         Thread t2 = new Thread(new RunnableThread(), "t2");
52         Thread t3 = new Thread(new RunnableThread(), "t3");
53         Thread t4 = new Thread(new RunnableThread(), "t4");
54         Thread t5 = new Thread(new RunnableThread(), "t5");
55         System.out.println("Starting Runnable threads");
56         t1.start();
57         t2.start();
58         t3.start();
59         t4.start();
60         t5.start();
61         System.out.println("Runnable Threads has been started");
62         Thread t6 = new MyThread("t6");
63         Thread t7 = new MyThread("t7");
64         Thread t8 = new MyThread("t8");
65         Thread t9 = new MyThread("t9");
66         Thread t10 = new MyThread("t10");
67         System.out.println("Starting MyThreads");
68         t6.start();
69         t7.start();
70         t8.start();
71         t9.start();
72         t10.start();
73         System.out.println("MyThreads has been started");
74     }
75 }
76 }
```

2. Type the following code and check the output.

```
1  import java.awt.Graphics;
2  import java.awt.Font;
3  import java.util.Date;
4  import java.applet.*;
5  /*
6   <applet code="ClockApplet" width="250" height="200">
7   </applet> */
8  public class ClockApplet extends Applet implements Runnable {
9
10     Font theFont = new Font("TimesRoman", Font.BOLD, 24);
11     Date theDate;
12     Thread runner;
13
14     public void start() {
15         if (runner == null) {
16             runner = new Thread(this);
17             runner.start();
18         }
19     }
20
21     public void stop() {
22         if (runner != null) {
23             runner.stop();
24             runner = null;
25         }
26     }
27
28     public void run() {
29         while (true) {
30             theDate = new Date();
31             repaint();
32             try {
33                 Thread.sleep(1000);
34             } catch (InterruptedException e) {
35             }
36         }
37     }
38
39     public void paint(Graphics g) {
40         g.setFont(theFont);
41         g.drawString(theDate.toString(), 10, 50);
42     }
43 }
44
```

Key Differences Between Thread and Runnable in Java

1. Each thread created by extending the Thread class creates a unique object for it and get associated with that object. On the other hand, each thread created by implementing a Runnable interface share the same runnable instance.
2. As each thread is associated with a unique object when created by extending Thread class, more memory is required. On the other hand, each thread created by implementing Runnable interface shares same object space hence, it requires less memory.
3. If you extend the Thread class then further, you can inherit any other class as Java do not allow multiple inheritance whereas, implementing Runnable still provide a chance for a class to inherit any other class.
4. One must extend a Thread class only if it has to override or specialise some other methods of Thread class. You must implement a Runnable interface if you only want to specialise run method only.
5. Extending the Thread class introduces tight coupling in the code as the code of Thread and job of thread is contained by the same class. On the other hand, Implementing Runnable interface introduces loose coupling in the code as the code of Thread is seprate from the job assigned to the thread.

2 Assignments

1. (a) Write a Java program to implement four threads using **Thread** class. Initialize each thread with a name: “T1”, “T2”, “T3”, and “T4”. Each thread prints the numbers from 1 to 10. Add a sleep time of 1 second after printing each number. Print name, state, and priority of each thread.
(b) Assign the following priorities to each thread and repeat the program.

Thread Name	Priority
<i>T1</i>	<i>MAX_PRIORITY</i>
<i>T2</i>	<i>MIN_PRIORITY</i>
<i>T3</i>	<i>NORM_PRIORITY</i>
<i>T4</i>	Any number between 1 to 10

2. (a) Write a Java program to implement four threads using **Runnable** Interface. Initialize each thread with a name: “T1”, “T2”, “T3”, and “T4”. Each thread prints the numbers from 1 to 10. Add a sleep time of 1 second after printing each number. Print name, state, and priority of each thread.
(b) Assign the following priorities to each thread and repeat the program.

Thread Name	Priority
<i>T1</i>	<i>MAX_PRIORITY</i>
<i>T2</i>	<i>MIN_PRIORITY</i>
<i>T3</i>	<i>NORM_PRIORITY</i>
<i>T4</i>	Any number between 1 to 10

3. Write a Java program to print a string in an applet. Initially, the string should be printed in red color. Change the color of the string after each 1 second.

Hint: Apply multithreading.

4. Write a Java applet program to implement a moving box which will change its fill color over time. Your name should be written inside the box. The interval of changing the position of the box and changing the fill color should be different. The color of the box should be cyan and red. The color of the text should be black. The motion of the box should be horizontal.

Hint: Apply multithreading.