Learning Java 9 : Databases and Multithreading in Java

User Assignment-4

Problem Statement:

1. Write a java program to create a thread by using Thread class and also by using the Runnable interface and display the details of thread like, thread name, id, priority, its phase and other details.
2. Write a java program to create three different threads, with first thread displaying numbers from 101 to 200, second from 201 to 300 and third one from 301 to 400 and verify that all the threads are running simultaneously or not.

```java
package com.question4.second;

class first extends Thread{
    int i;
    public void run() {
        for(i=101;i<=200;i++) {
            System.out.println("first : "+i);
            try {
                Thread.sleep(500);
            }
            catch(InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```java
package com.question4;

public class ThreadClass_RunnableThread{
    public static void main(String[] args) {
        int id;
        System.out.println("main name: "+Thread.currentThread().getName());
        System.out.println("main id: "+Thread.currentThread().getId());
        System.out.println("main priority:
"+Thread.currentThread().getPriority());
        Threadclass t = new Threadclass();
        t.start();
        System.out.println("main method name:
"+Thread.currentThread().getName());
        System.out.println("main method id: "+Thread.currentThread().getId());
        System.out.println("main method priority:
"+Thread.currentThread().getPriority());
        Runnable r = new Runnableinterface();
        Thread t1 = new Thread(r);
        t1.start();
    }
}
```

```java
package com.question4;
import java.util.*;
class Threadclass extends Thread{
    private String name;
    public synchronized void run() {
        System.out.println("executing run method from Thread (Thread Name):
"+Thread.currentThread().getName());
        System.out.println("executing run method from Thread (Thread Id):
"+Thread.currentThread().getId());
        System.out.println("executing run method from Thread (Thread
Priority): "+Thread.currentThread().getPriority());
        System.out.println("Phase of Thread in Thread class:
"+Thread.currentThread().getState());
    }
}
class Runnableinterface implements Runnable{
    public synchronized void run() {
        System.out.println("executing run method from Runnable:
"+Thread.currentThread().getName());
        System.out.println("executing run method from Runnable:
"+Thread.currentThread().getId());
        System.out.println("executing run method from Runnable:
"+Thread.currentThread().getPriority());
        System.out.println("Phase of Thread in Runnable class:
"+Thread.currentThread().getState());
    }
}
```

```java
package com.question4.second;

public class Three_thread_running_simultaneously {
    public static void main(String arg[]) throws InterruptedException
    {
        first f=new first();
        f.start();

        Thread.sleep(10);
        two t=new two();
        t.start();
        Thread.sleep(10);

        three t1=new three();
        t1.start();



    }

}
```

```java
package com.question4.second;

class three extends Thread{
    int i;
    public void run()
    {
        for(i=301;i<=400;i++)
        {
            System.out.println("three : "+i);
        try {
            Thread.sleep(500);
            } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            }
        }
    }
}
```

```java
package com.question4.second;

class two extends Thread{
    int i;
    public void run() {
        for(i = 201;i<=300;i++) {
            System.out.println("two : "+i);
            try {
                Thread.sleep(500);
            }
            catch(InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```