Experiment Number: 6 (Multithreading)

Name	**Shreya Shetty**
UID	**2019140059**
Class	**TE IT**
Batch	**D**
Subject	**OOP Lab**

Aim: There are 3 threads present. The 1st thread will make any random number x and sleeps for 3 seconds time. Now if the number x is even then the 2nd thread will find the factorial of the number. If x is odd then the 3rd thread will find the sum of all factors of the number. Write a program in java and display the output in a systematic format after every iteration perform the following activities with thread :

1. get current thread name
2. set the thread name
3. pause a thread for 1.5 seconds
4. Check weather the thread is running state or not
5. Check and make sure that your new thread completes before main thread completes

Program:

```
//Shreya Shetty TE IT 2019140059 Batch D

package oopExp6;
import java.util.Random;

class RandomNumberThread extends Thread {
    public void run() {
        Random random = new Random();
        for (int i = 0; i < 10; i++)
        {
            int randomInteger = random.nextInt(25);
            System.out.println("\nRandom Integer generated :  " +
randomInteger);
            try {
                System.out.println("Random Thread is sleeping");
                Thread.sleep(3000);
            }
            catch (InterruptedException ex) {
                System.out.println(ex);
            }
            if((randomInteger%2) == 0) {
                FactorialThread sThread = new FactorialThread(randomInteger);
                System.out.print("Name of current thread :  "+
sThread.getName()+"\n");
                sThread.setName("Factorial-Thread");
```

```
                System.out.println("After changing name of current thread :
"+sThread.getName());
                sThread.start();
                System.out.println("Current state of thread
"+sThread.getName()+" is "+sThread.getState());
                try {
                    System.out.println("Current thread is sleeping");
                    Thread.sleep(1500);
                }
                catch (InterruptedException ex) {
                    System.out.println(ex);
                }
                System.out.println("Current state of thread
"+sThread.getName()+" is "+sThread.getState());
            }
            else {
                FactorSumThread cThread = new FactorSumThread(randomInteger);
                System.out.print("Name of current thread :  "+
cThread.getName()+"\n");
                cThread.setName("Factors-Sum-Thread");

                System.out.println("After changing name of current thread :
"+cThread.getName());
                // System.out.println("Current state of thread is
"+cThread.getState());
                cThread.start();
                System.out.println("Current state of thread
"+cThread.getName()+" is "+cThread.getState());
                try {
                    System.out.println("Current thread is sleeping");
                    Thread.sleep(1500);
                }
                catch (InterruptedException ex) {
                    System.out.println(ex);
                }
                System.out.println("Current state of thread
"+cThread.getName()+" is "+cThread.getState());

            }
        }

    }
}

class FactorialThread extends Thread
{
    int number;
    FactorialThread(int randomNumbern)
    {
        number = randomNumbern;
```

```
    }
    public void run() {
        long factor=1;
        for(int i=1;i<=number;i++)
        {
            factor=factor*i;
        }
        System.out.println("Factorial of "+number+" is :  "+factor);
    }
}

class FactorSumThread extends Thread {
    int number;
    FactorSumThread(int randomNumber) {
        number = randomNumber;
    }
    public void run() {
        int fact_sum = 1;
        for (int i = 2; i <= Math.sqrt(number); i++)
        {
            int  curr_sum = 1;
            int curr_term = 1;
            while (number % i == 0)
            {
                number/=i;
                curr_term *= i;
                curr_sum += curr_term;
            }
            fact_sum *= curr_sum;
        }
        if (number > 2)
            fact_sum *= (1 + number);

        System.out.println("Sum of factors of " + number + " is " + fact_sum);
    }
}

public class MultiThreading {
    public static void main(String args[]) {
        RandomNumberThread rnThread = new RandomNumberThread();
        System.out.print("Name of current thread :  "+
rnThread.getName()+"\n");
        rnThread.setName("Random-Thread");
        rnThread.start();
        System.out.println("After changing name of current thread :
"+rnThread.getName());
        System.out.println("Current state of thread "+rnThread.getName()+" is
"+rnThread.getState());
    }
}
```

```
}
```

Output:

```
PS D:\PROJECT_AND_CODES>  & 'C:\Program Files\Java\jdk-11.0.12\bin\java.exe' '-cp' 'D:\PROJECT_AND_CODES\Java' 'oopExp6.MultiThreading'
Name of current thread :  Thread-0
After changing name of current thread : Random-Thread
Current state of thread Random-Thread is RUNNABLE

Random Integer generated :  6
Random Thread is sleeping
Name of current thread :  Thread-1
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 6 is :  720
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  10
Random Thread is sleeping
Name of current thread :  Thread-2
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 10 is :  3628800
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  0
Random Thread is sleeping
Name of current thread :  Thread-3
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 0 is :  1
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  7
Random Thread is sleeping
Name of current thread :  Thread-4
After changing name of current thread : Factors-Sum-Thread
Current state of thread Factors-Sum-Thread is RUNNABLE
Current thread is sleeping
Sum of factors of 7 is 8
Current state of thread Factors-Sum-Thread is TERMINATED
```

```
Random Integer generated :  2
Random Thread is sleeping
Name of current thread :  Thread-5
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 2 is :  2
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  18
Random Thread is sleeping
Name of current thread :  Thread-6
After changing name of current thread : Factorial-Thread

Random Integer generated :  24
Random Thread is sleeping
Name of current thread :  Thread-7
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 24 is :  -7835185981329244160
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  14
Random Thread is sleeping
Name of current thread :  Thread-8
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 14 is :  87178291200
Current state of thread Factorial-Thread is TERMINATED

Random Integer generated :  4
Random Thread is sleeping
Name of current thread :  Thread-9
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 4 is :  24
Current state of thread Factorial-Thread is TERMINATED
```

```
Random Integer generated :  4
Random Thread is sleeping
Name of current thread :  Thread-10
After changing name of current thread : Factorial-Thread
Current state of thread Factorial-Thread is RUNNABLE
Current thread is sleeping
Factorial of 4 is :  24
Current state of thread Factorial-Thread is TERMINATED
PS D:\PROJECT_AND_CODES>
```