LAB # 3

Introduction to Concurrency

OBJECTIVE

Understanding and implementing the concept of concurrency through different mechanisms of multithreading.

Lab Task:

1. Implement the following program on eclipse IDE and answer the following questions:
   - How many threads are running?
   - How many tasks are running?
   - If more tasks are added than what will be the impact on number of threads?
   - Explain the flow of program:

```
class Main extends Thread{
       public void run(){
         System.out.println("task one");
       }
       public static void main(String args[]){
        Main t1=new Main();
        Main t2=new Main();
        Main t3=new Main();
        t1.start();
        t2.start();
        t3.start();
       }
      }
```

ANSWER:

- 3 threads are running.

Each time you call `.start()`, a **new thread** is created.

`t1.start()` → starts Thread 1

`t2.start()` → starts Thread 2

`t3.start()` → starts Thread 3

- 3 tasks are running.

  Each thread executes the **same task** defined in the `run()` method

- If you create **more task objects** (like `t4`, `t5`, etc.) and call `.start()` on them, then **more threads will be created**.
  Each `.start()` call → **creates one new thread**.
  So if you add more tasks → the number of threads **increases linearly**.

- **Main thread starts** the execution.

  In `main()`:

  Three thread objects are created: `t1`, `t2`, `t3`.

  `t1.start()` is called → it internally calls the `run()` method of `t1` in a **new thread**.

  Similarly, `t2.start()` and `t3.start()` each start their own new threads.

  Now, **3 threads run concurrently**, each printing `"task one"`.

The order of output may **change each time** you run the program because threads execute **asynchronously**.

OUTPUT:

```
task one
task one
task one
```

2. With the help of threading print two tables concurrently, print one table number of student roll number e.g. 2019-SE-092 and second number should be date of birth e.g. 05-April.

CODE:

```
    public void run() {
        System.out.println("\nVerifying Date of Birth Entries:");
        for (int i = 1; i <= 10; i++) {
            System.out.println("Processing DOB entry #" + i);
            try {
                Thread.sleep(400);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Main {
    public static void main(String[] args) {
        int startRoll = 42;

        Thread t1 = new Thread(new RollNumberPrinter("Roll Number Generator", startRoll));
        Thread t2 = new Thread(new DOBPrinter());

        t1.start();
        t2.start();

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

        System.out.println("\nAll roll numbers with random DOBs processed concurrently.");
    }
}

```

OUTPUT:

```
DOB Table ? 5 x 1 = 5
Roll Number Table ? 42 x 1 = 42
DOB Table ? 5 x 2 = 10
Roll Number Table ? 42 x 2 = 84
DOB Table ? 5 x 3 = 15
Roll Number Table ? 42 x 3 = 126
DOB Table ? 5 x 4 = 20
Roll Number Table ? 42 x 4 = 168
DOB Table ? 5 x 5 = 25
Roll Number Table ? 42 x 5 = 210
DOB Table ? 5 x 6 = 30
Roll Number Table ? 42 x 6 = 252
DOB Table ? 5 x 7 = 35
Roll Number Table ? 42 x 7 = 294
DOB Table ? 5 x 8 = 40
Roll Number Table ? 42 x 8 = 336
DOB Table ? 5 x 9 = 45
Roll Number Table ? 42 x 9 = 378
DOB Table ? 5 x 10 = 50
Roll Number Table ? 42 x 10 = 420
```