Aima khan
2023F-BSE-064

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:

```
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();
      }
     }
```

- How many threads are running?

The `Main` class extends `Thread`, and three objects (`t1`, `t2`, and `t3`) are create d. When we call `.start()` on each object, **a new thread is created and started** for each one. **So, there are 3 threads running concurrently**, plus the **main thread** that executes the `main()` method.

- How many tasks are running?

Each thread executes the `run()` method once. The `run()` method prints `"task one"`.Therefore, **3 tasks** are running — one task per thread.

- If more tasks are added than what will be the impact on number of threads?

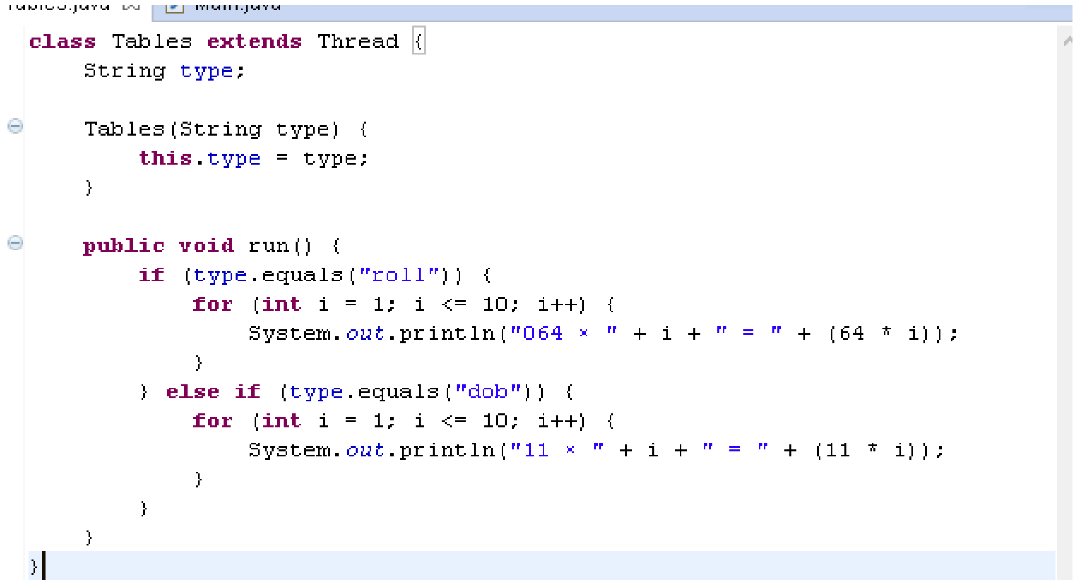
- **If more tasks are added:**
  Every time you create a new thread object and call `.start()`, a **new thread** is created. So, adding more tasks increases the number of threads. Too many threads can slow performance.

- Explain the flow of program:

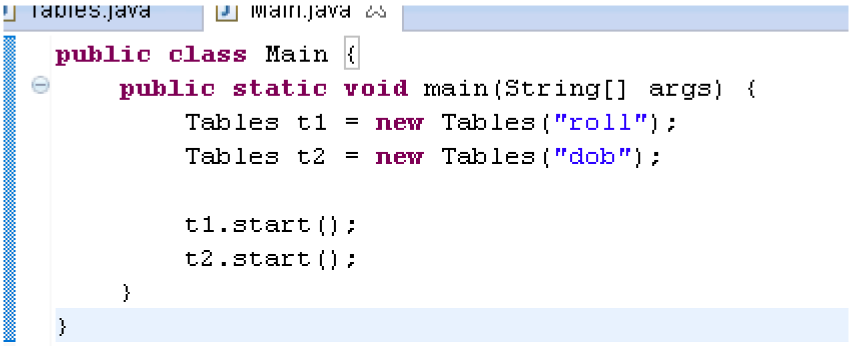
The program starts with the **main thread**, executing the `main()` method.Three `Main` objects (`t1`, `t2`, `t3`) are created — but **no threads are running yet**.When `t1.start()` is called:A new thread is created.That thread calls the `run()` method and prints `"task one"`.Similarly, `t2.start()` and `t3.start()` create two more threads, each executing `run()`.All three threads run **concurrently**, so the order of `"task one"` prints is **unpredictable**

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:

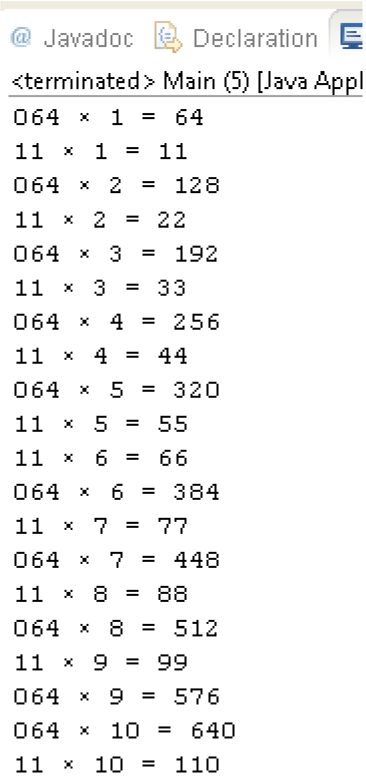


```
class Tables extends Thread {
    String type;

    Tables(String type) {
        this.type = type;
    }

    public void run() {
        if (type.equals("roll")) {
            for (int i = 1; i <= 10; i++) {
                System.out.println("064 × " + i + " = " + (64 * i));
            }
        } else if (type.equals("dob")) {
            for (int i = 1; i <= 10; i++) {
                System.out.println("11 × " + i + " = " + (11 * i));
            }
        }
    }
}
```



```
public class Main {
    public static void main(String[] args) {
        Tables t1 = new Tables("roll");
        Tables t2 = new Tables("dob");

        t1.start();
        t2.start();
    }
}
```

OUTPUT:



```
<terminated> Main (5) [Java Appl
064 × 1 = 64
11 × 1 = 11
064 × 2 = 128
11 × 2 = 22
064 × 3 = 192
11 × 3 = 33
064 × 4 = 256
11 × 4 = 44
064 × 5 = 320
11 × 5 = 55
11 × 6 = 66
064 × 6 = 384
11 × 7 = 77
064 × 7 = 448
11 × 8 = 88
064 × 8 = 512
11 × 9 = 99
064 × 9 = 576
064 × 10 = 640
11 × 10 = 110
```