

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?

4 threads in total 1 main thread and 3 child threads (t1, t2, t3).

How many tasks are running?

3 tasks each thread executes the `run()` method once.

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

Each new task started with `.start()` creates a new thread, so the number of threads increases.

```

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();
        Main t4=new Main();

        t1.start();
        t2.start(); |
        t3.start();
        t4.start();
    }
}

```

Explain the flow of program:

The main thread starts and creates three `Main` thread objects.

Calling `start()` on each creates a new thread.

Each thread runs the `run()` method concurrently and prints "task one".

Output order may vary because threads run simultaneously.

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.

```

1  class TableThread extends Thread {
2      int num;
3      String name;
4
5      TableThread(int num, String name) {
6          this.num = num;
7          this.name = name;
8      }
9
10     public void run() {
11         System.out.println("Table of " + num + " (" + name + ")");
12         for (int i = 1; i <= 10; i++) {
13             System.out.println(name + ": " + num + " x " + i + " = " + (num * i));
14             try {
15                 Thread.sleep(100);
16             } catch (InterruptedException e) {
17                 System.out.println(e);
18             }
19         }
20         System.out.println("End of " + name + " table\n");
21     }
22
23     public static void main(String[] args) {
24         TableThread rollTable = new TableThread(319, "Roll No 319");
25         TableThread dobTable = new TableThread(18, "DOB 18");
26
27         rollTable.start();
28         dobTable.start();
29     }
30 }
```

OUTPUT:

Table of 18 (DOB 18)
 Table of 229 (Roll No 229)

Roll No 229: 229 x 1 = 229
 DOB 18: 18 x 1 = 18

Roll No 229: 229 x 2 = 458
 DOB 18: 18 x 2 = 36

Roll No 229: 229 x 3 = 687
 DOB 18: 18 x 3 = 54

Roll No 229: 229 x 4 = 916
 DOB 18: 18 x 4 = 72

Roll No 229: 229 x 5 = 1145
 DOB 18: 18 x 5 = 90

Roll No 229: 229 x 6 = 1374
 DOB 18: 18 x 6 = 108

Roll No 229: 229 x 7 = 1603
 DOB 18: 18 x 7 = 126

Roll No 229: 229 x 8 = 1832
 DOB 18: 18 x 8 = 144

Roll No 229: 229 x 9 = 2061
 DOB 18: 18 x 9 = 162

Roll No 229: 229 x 10 = 2290
 DOB 18: 18 x 10 = 180

End of DOB 18 table
 End of Roll No 229 table