# Distributed and Cloud Computing Laboratory

**B.Tech. 6th Semester**


**Name** : K Srikanth

**Roll Number** : 17ETCS002124

**Department** : Computer Science and Engineering



## Faculty of Engineering & Technology

**Ramaiah University of Applied Sciences**

Name: K Srikanth Registration Number: 17ETCS002124

# Ramaiah University of Applied Sciences

Private University Established in Karnataka State by Act No. 15 of 2013

| | |
|---|---|
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Year/Semester | 2nd Year / 6th Semester |
| Name of the Laboratory | Distributed and Cloud Computing Laboratory |
| Laboratory Code | 19CSL316A |

Name: K Srikanth Registration Number: 17ETCS002124

# Laboratory 1

Title of the Laboratory Exercise: Multithreaded Programs in Java

1. **Introduction and Purpose of Experiment**

   Multithreading is the ability of a single core or a multi-core processor to execute multiple threads concurrently, supported by Java run time system. By solving this students will be able to manipulate multiple threads in a Java program.

   Aim and Objectives

   Aim

   - To develop Java multithreaded programs

2. **Experimental Procedure**

   i. Analyse the problem statement
   ii. Design an algorithm for the given problem statement and develop a flowchart/pseudo-code
   iii. Implement the algorithm in Java language
   iv. Compile the Java program
   v. Test the implemented program
   vi. Document the Results
   vii. Analyse and discuss the outcomes of your experiment

3. **Questions**

   Implement the following:

- Create two Java threads and display Hello World by them
- Create four Java threads and display the results of addition, subtraction, multiplication and division of two numbers by each thread.

4. **Calculations/Computations/Algorithms**

- **Create two Java threads and display Hello World by them**

## Algorithm

1. Start
2. Create a Class **"HelloWorldMethod"** extends to **Threads**
3. Create a Run Function inside the Class
    a. **Print( "Hello World")**
4. Create the Object of the Class **"HelloWorldMethod"**
5. Run the Thread using **"Object_Name.Start()"**
6. Join the Thread using **"Object_Name.Join()"**

- **Create four Java threads and display the results of addition, subtraction, multiplication and division of two numbers by each thread.**

## Algorithm

1. Start
2. Create a Class **"Addition","Subtraction","Multiplication","Division"** extends to **Threads**
3. Create a Run Function inside these Class
    a. **Addition ( X + Y ) → 1st Class**
    b. **Subtraction ( X – Y ) → 2nd Class**
    c. **Multiplication ( X * Y ) → 3rd Class**
    d. **Division ( X / Y ) → 4th Class**
4. Create the Object of the All Class **"Addition","Subtraction","Multiplication","Division"**
5. Run the Thread using **"Object_Name.Start()"**
6. Join the Thread using **"Object_Name.Join()"**

## 5. Presentation of Results

- **Create two Java threads and display Hello World by them**

### Code

```
class HelloWorldMethod extends Thread {
    public void run() {
            System.out.println("Hello World Created by Thread " + Thread.currentThread().getId() + ".");
    }
}
```

Figure 1 Java Program for Printing "Hello World " Thread using a Thread Class “Hello World Method”

- **Create four Java threads and display** the results of addition, subtraction, multiplication and division of two numbers by each thread.

### Code

### Addition Class

```
50  class Addition extends Thread {
51      int a, b;
52
53      Addition(int x, int y) {
54          a = x;
55          b = y;
56      }
57
58      public void run() {
59          int add = a + b;
60          System.out.println("The Sum of " + a + " and " + b + " is " + add);
61      }
62  }
```

Figure 2 Java Program for Addition of Two Numbers using a Thread Class “Addition Method”

### Subtraction Code

```
64  class Subtraction extends Thread {
65      int a, b;
66
67      Subtraction(int x, int y) {
68          a = x;
69          b = y;
70      }
71
72      public void run() {
73          int sub = a - b;
74          System.out.println("The Subtraction of " + a + " and " + b + " is " + sub);
75      }
76  }
```

Figure 3 Java Program for Subtraction of Two Numbers using a Thread Class “Subtraction Method”

## Multiplication Code

```
78  class Multiply extends Thread {
79      int a, b;
80
81      Multiply(int x, int y) {
82          a = x;
83          b = y;
84      }
85
86      public void run() {
87          int mull = a * b;
88          System.out.println("The Multiplication of " + a + " and " + b + " is " + mull);
89      }
90  }
```

Figure 4 Java Program for Multiplication of Two Numbers using a Thread Class “Multiply Method”

## Division Code

```
92  class Division extends Thread {
93      int a, b;
94
95      Division(int x, int y) {
96          a = x;
97          b = y;
98      }
99
100     public void run() {
101         int div = a / b;
102         System.out.println("The Division of " + a + " and " + b + " is " + div);
103     }
104 }
```

Figure 5 Java Program for Division of Two Numbers using a Thread Class “Division Method”

## Main Function

```java
import java.util.Scanner;
public class App {
    Run | Debug
    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.println("");
        System.out.println("********** K Srikanth 17ETCS002124 **********");
        System.out.println("********** Lab 1 **********");
        System.out.println("");
        System.out.println("********** Question 1 **********");
        int n = 2; // Number of threads
        for (int i = 0; i < n; i++) {
            HelloWorldMethod Hello = new HelloWorldMethod();
            Hello.start(); // Two Threads for Hello World
            Hello.join(); // Joining the Threads
        }
        System.out.println("");
        System.out.println("********** Question 2 **********");
        System.out.println("********** Calculator **********");
        System.out.println("Enter the 1st Number : ");
        int x = input.nextInt();
        System.out.println("Enter the 2nd Number : ");
        int y = input.nextInt();
        Addition add = new Addition(x, y);
        Subtraction sub = new Subtraction(x, y);
        Multiply mull = new Multiply(x, y);
        Division div = new Division(x, y);
        add.start(); // Thread 1
        sub.start(); // Thread 2
        mull.start(); // Thread 3
        div.start(); // Thread 4
        add.join(); // Joining Thread 1
        sub.join(); // Joining Thread 2
        mull.join(); // Joining Thread 3
        div.join(); // Joining Thread 4
        input.close(); // Done Taking Input
    }
}
```

Figure 6 Java Program for Main Class

## Output

```
********** K Srikanth 17ETCS002124 **********
********** Lab 1 **********

********** Question 1 **********
Hello World Created by Thread 12.
Hello World Created by Thread 13.

********** Question 2 **********
********** Calculator **********
Enter the 1st Number :
2
Enter the 2nd Number :
3
The Sum of 2 and 3 is 5
The Subtraction of 2 and 3 is -1
The Multiplication of 2 and 3 is 6
The Division of 2 and 3 is 0
```

Figure 7 Java Program Output of the Above Program

## 6. Analysis and Discussions

### Introduction

**Java is a multi-threaded programming language** which means we can develop multi-threaded program using this language. A multi-threaded program contains **two or more parts that can run concurrently** and each part **can handle a different task at the same time making optimal use of the available resources** specially when your computer has multiple CPUs. We can achieve this using multiple processes share common processing **resources such as a CPU** and it also allows the **user to write in a way where multiple activities can proceed concurrently in the same program.**

### Creating Thread

To create a thread in java first we have to declare **a class** which then extends to the **Thread Class** Now that you created a some random class to **run your threads**. The next step is that you have to **write the code** whether what would your thread class would be doing so for **example let's take a scenario that I'm creating a thread class to print 2 "Hello Worlds"** so now we have to create a **method that can run our code using threads** so that method is **run() inside this method you** write your logic… for now **I'm printing "hello world" using this method.**

### Running Thread

Now that we have **created our thread class to print "hello world"** now what we do is we **make an object of that class** inside the main function then to run the thread we just call another method called **"start()" with your Object Name. Now** what is method does it that it will **trigger your thread clas**s and **look for a run() method** then it will start executing after it gets triggered and it starts executing.

### Joining Thread

Now after we are done with **our thread method now we have to wait for other threads to complete their respective processes** that they were assigned to so at the end we **join all the threads** using **"Join()" Method with your Object Name.**

Name: K Srikanth Registration Number: 17ETCS002124

**1. Limitations of Experiments**

Creation of threads can be minimal.

**2. Limitations of Results**

We are creating four threads to perform 4 different operations with the same data.. that can be reduced

**3. Learning happened**

Learned how to perform operations with threads in Java Programming Language.