**CIS 181 - Lab 4**

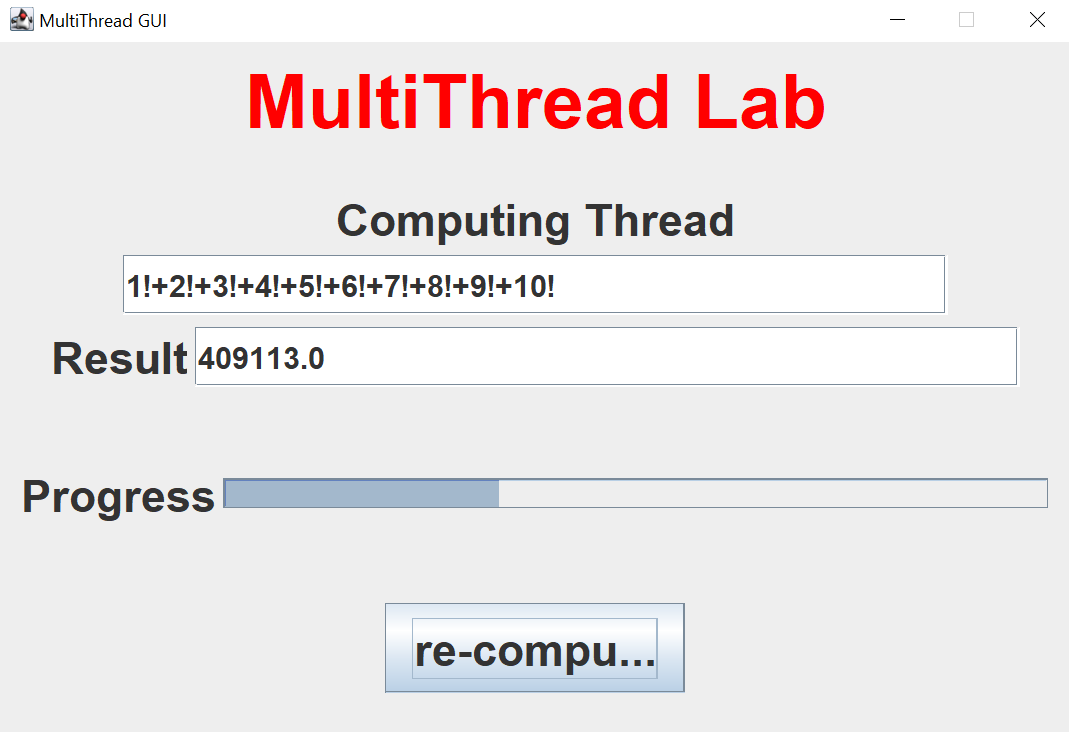
**Factorial Computing and Displaying Using 2 Threads**

## ****Objectives****

* To gain experience with developing multithreading programming in Java.
* To gain experience with using two types of creating threads*.*
* To enjoy developing concurrent software system.

**Description**

There are two thread used in this lab. One thread is responsible for computing factorial （1！+2！+3！……+30!）. Evaluating each factorial expression is a random number between 0.5 second to 1 second. The second thread reads the first thread’ computing results and progress, and displays progress bar in GUI. In summary, the first thread is responsible for computing factorial, and the second thread is responsible for displaying computing progress and current computing results. A example of GUI running is presented below.



You are required to (1) design a GUI to display progress and results correctly; (2) implement two types of creating thread using subclass of Thread class and Runnable interface; and (3) test your code.

There are three java files and their purposes.

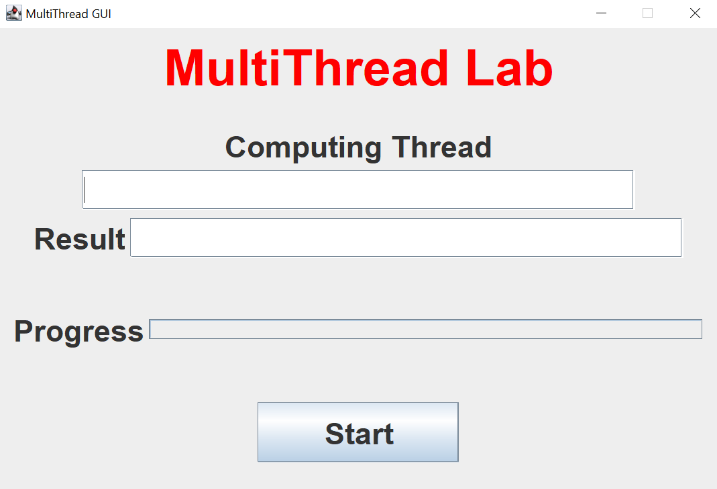
(1). “ThreadFrame.java”: GUI implementation

(2). “ComputeThread.java”: computing factorial

(3). “ReturnThread.java”: return computing results for displaying

**Exercises**

1. Download the following class source files:
   * ThreadFrame.java (**Note:** Don’t need to change any code)
   * ComputeThread.java (**Note:** Need to complete run function)
   * ReturnThread.java (**Note:** Need to complete run function)
2. In Eclipse, create a new Java project called "Lab 4 ", and import the above 3 files into a package called “Thread”.
3. Compile and run the program. Make sure it works. You can obtain the following GUI. But run() functions are missing, if you push button “Start”, there is no computing result and progress bar working. You need to add your code into run() functions in both ComputeThread and ReturnThread classes.



1. Modify the classes called *ComputeThread and ReturnThread*:
   * Add code into run() function in ComputThread class.
   * Add code into run() function in ReturnThread class.
   * Make those two threads run together.
2. Submit your completed program (ComputeThread.java and ReturnThread.java) to the myCourses!