# Tutorial 4 – Java Thread

## Instructions

1. Read the attached material about synchronization in Java programing.
2. Run the sample code (`UnsynchronizedCounterTest.java`) and see what would happen without "synchronized".

   (You may need to try many times to track the bug. Say "Bingo" if the bug happens.)
3. Write a program to find the integer in the range 1 to 200,000 that has the largest number of divisors. At the end of the program, output the elapsed time, the integer that has the largest number of divisors, and the number of divisors that it has.

   (*) Stop the program and check for infinite loop problem, or reduce the range, if it takes too long (for example, more than 1 minute).
4. Now write a program that uses multiple threads to solve the same problem, but for the range 1 to 200,000 (or less, if you don't have a fast computer). By using threads, your program will take less time to do the computation when it is run on a multiprocessor computer.

   a) To turn this into a multi-threaded program, we have to divide the integers between 1 and 200,000 into groups and assign each group of integers to a thread. In its run method, each thread finds the maximum number of divisors for integers in its assigned group.

   b) The only problem is combining the final results of all the threads.