Peterson's Algorithm in Java

On the course homepage, you will find source code for a Java program, Peterson.java. This program demonstrates a race condition, much like some of the sample programs we looked at in class. If you compile and run this program, you should see evidence of the race condition. The output should be zero, but it's not. In fact, the output varies from execution to execution.

We know that Peterson's solution is not guaranteed to work on a modern microprocessor. Let's see if it works in Java. Add Peterson's solution entry code and exit code around just the critical sections in each thread. Then, run your program (preferably on a multi-core system) and see if your Peterson's solution code seems to work.

To implement Peterson's solution, you'll need some more variables shared between the two threads (e.g., a pair of flags and a turn variable). Just like the sharedVar and running variables I've provided, you can create these as static fields in the Peterson class. You'll need to mark these variables as **volatile**. This tells the compiler that these variables may be modified by another thread, and it should write code that always goes to the memory location where these variables are stored whenever it needs to access them (rather than, say, temporarily storing the value in a CPU register).

Try out your program a few times, with the Peterson's solution code in place. Does it work (does it not deadlock and does it seem to fix the race condition)? On the second line of the source file, I've left a blank comment for you to write your answer. Just write the word “yes” or “no” to report whether this solution worked for you. I'm going to grade this exercise with a simple grading script, so be sure you just put the word **yes** or **no** in this comment and keep it on the second line of the source file, so my script can find it. Don’t remove the // at the start of the line. Otherwise, your code won’t compile. Also, be sure not to move this line elsewhere, or my grading script won’t be able to find it.

When your Peterson.java program is done (remember, it may or may not exhibit a race condition, so I'd rather not say “when it's working”) submit it using the exercise_08 assignment on Moodle.