R·I·T | Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences & Technology

ISTE-200 Java for Programmers

Name: ________________________________

08b – Threads – Part 1

Objectives

This assignment will have you working with threads, and the progress bar. Remember, you can use any resource you want for completing the assignments. What is not allowed is anything that will violate the RIT or IST Academic Dishonesty policy such as copying code. Working in pairs or teams of two or three is encouraged.

NOTE: Based on speeds of the computers, and number of processors, the following assignment my work different on different computers. Contact the instructor or TA with questions.

Part 1a – Write a simple Thread

Write a NON-GUI class, ThreadPe.java that has a main, which calls the ThreadPe constructor. The constructor creates two instances of the inner class ThreadPeInner that extends the thread class. When instantiating the ThreadPeInner constructor, pick a name for each thread and pass the name of the thread to the constructor. The inner class's run method prints, "This ran thread" followed by the thread name.
After the ThreadPe constructor instantiates and starts both threads, have it print "Program finished". Write the output below.

```
This ran thread First thread
This ran thread Second thread
Program finished
```

Part 1b – not so fast, slow down and yield()!

At the beginning of the run() method, place a yield(). What is the output now? (With the multiple CPU's this may not change any results.)

Program finished

This ran thread First thread

This ran thread Second thread

Part 1c – Calculate something

If everything went as expected, the "Program finished" came out first.
With a dual core, it possibly didn't come out first.
To the ThreadPe class, add an int attribute that will be a counter, initialize it to zero.
To the "Program finished", print out this out as the "Program finished, count = ", and the counter.
To the end of the run() method in ThreadPeInner, add one to the counter.
Now what is the output?

Program finished, count = 0

This ran thread First thread

This ran thread Second thread

Why was the counter zero?

Part 1d – Wait for the computation to finish, then join the party

To the ThreadPe constructor add code between the start and print, that makes that code wait until each of the threads has completed their execution. You may not use the sleep method.
What method did you use? join()
What was the output now?

This ran thread First thread

Order can flip

This ran thread Second thread

Program finished, count = 2

Have the Instructor / TA check your work at this point.

Instructor / TA: ____________________