

Parallel Programming in Java

• Java supports multithreaded (shared-memory parallel) programming as part of the language — synchronized keyword, wait and notify methods of Object class, Thread class. Programs that use the GUI classes (AWT or Swing) multithreaded under the hood. Justification probably has more to do with hiding latency than HPC, but still useful, and versions 5.0 and beyond includes much useful library stuff.

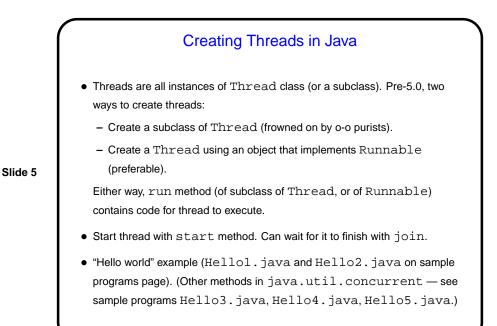
Slide 3

 Java also provides support for forms of distributed-memory programming, through library classes for networking, I/O (java.nio), and Remote Method Invocation (RMI).

What Does A Multithreaded Java Program Look Like?

- Easy answer: Like a regular Java program. (In fact, any program with a GUI ...)
- Programming model: All threads share a common address space.
 Programmer is responsible for creating threads, providing synchronization, etc.

Slide 4

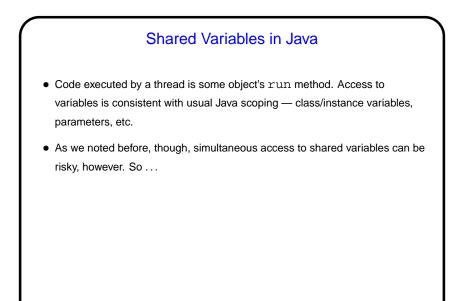




• Most of you probably use Eclipse to write Java programs. You can do that for this course too, but for this course you will likely prefer to run them from the command line (since you need to supply environment variables that will vary from run to run). Command to use is java, followed by class name and any arguments. (If class files are not in current directory, specify where they are with -classpath.)

You can also write them using your favorite text editor compile from the command line. Command to compile is javac.

Slide 6



Slide 7

