Slide 1

Administrivia

- Reminder: Homework 7 design due today, code next Tuesday. Homework 8 design due next Tuesday also.
- Binary-I/O example from Tuesday revised to get input from standard input (using a `Scanner`) rather than command-line arguments.

Slide 2

Multithreading Basics

- What's a thread? Conceptually, a sequence of steps executed one at a time.
- "Multithreading" — similar to operating system's "multitasking" — execute more than one thread (application) in effect at the same time. Why?
  - For better performance, if there's more than one CPU or to "hide latency".
  - Because it's a good mental model — e.g., for GUIs.
- Threads can share variables — useful, but risks "race conditions". For this and other reasons, sometimes want one thread to wait for another to do something.

Slide 3

Threads in Java

- `Thread` class provides basic functionality. To start a new thread, make a `Thread` object and call its `start` method. Two choices:
  - Create a `Thread` with an object that implements `Runnable` — `run` method has code to execute.
  - Define a subclass of `Thread` that has a `run` method with code to execute.
- Interthread interaction based on "monitors" (see textbooks on operating systems, parallel programming).
  - Every object (and every class) has a lock.
  - `synchronized` methods must acquire lock — so only one at a time can run.
  - `wait` gives up the lock and sleeps; `notify` and `notifyAll` wake up one/all sleeping thread(s).

Slide 4

Threads in Java, Continued

- Other useful methods:
  - `Thread.sleep` makes current thread sleep for some interval.
  - `t.join` wait for `Thread t` to finish.
  - `t.interrupt` interrupts `Thread t` (which can check whether it has been interrupted with `isInterrupted` — safe/approved way for one thread to stop another.
- Can set thread priorities — sometimes useful, but not a substitute for proper synchronization.
- Lots of new threads-related stuff in Java 1.5 / 5.0 (`java.util.concurrent` package).

Slide 5

Examples

- Formerly many uses for multithreading in GUIs (e.g., animation), but now most can be accomplished with new features of GUI class (e.g., timers). Still useful, however, if you want something that might take a while to execute in the background. Examples ...
- Examples of multithreading for performance, multithreading with `wait` and `notify` ...

Slide 6

Minute Essay

- None — quiz.