Administrivia

- Code for sorted binary tree class will be on Web soon. Could be interesting to try your own version — without recursion?
- Grades for Homework 4 mailed. Homework 5 being graded. If you're struggling with something, ask for help!
- Homework 7 design due today at 11:59pm. Questions?

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.
  - 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.

Minute Essay From Last Lecture

- Show what a heap looks like after inserting 5, 4, -1, 10, 6, 20. (Let's do this at the board …)

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.
- Inter-thread interaction based on “monitors” (see o/s or parallel-programming textbooks):
  - 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).
Threads in Java, Continued

- Thread.sleep to make current thread sleep for some interval.
- Can set thread priorities — sometimes useful, but not a substitute for proper synchronization.

Examples

- Example of multithreading for performance …
- Example of multithreading for GUI …
- Example of multithreading with wait and notify …

Minute Essay

- Are there features you want to add to your game, but you have no idea how?