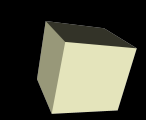
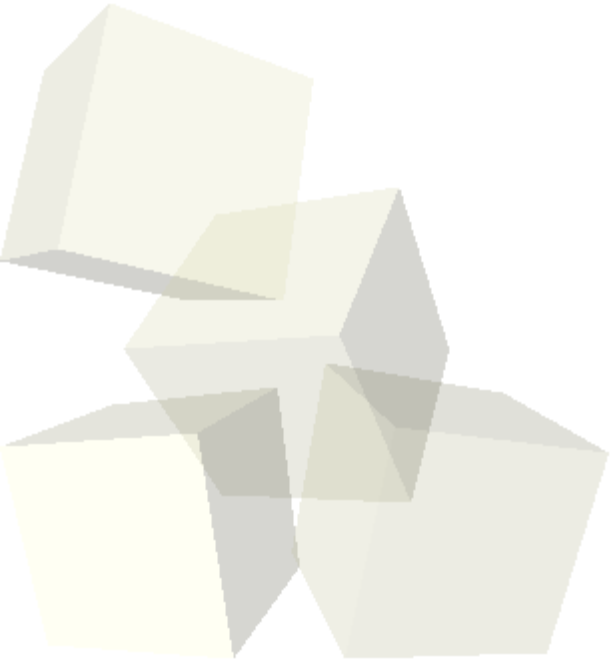
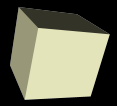


Threads

2/8/2007





Opening Discussion

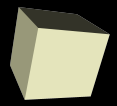
- When would you write your own sort? If you know something about the data or it is a very small set of data. Mergesort has a lot of overhead that hurts you when N is small.
- Do you have any questions about the reading?
- Do you have any questions about the assignment?
- What are threads? Why do you care about them?
- How do you make multiple threads in Java?
- What are some difficulties you run into with threads?
- What can you do in Java to get around these difficulties?

Thread Coding

- The sorting code we wrote last time can provide a great test for threading. In this case we want to use our slow sorts so that we can actually time how long it takes them to run.
- I want you to add to your main some code that will create N (where N is an int variable) arrays of Doubles of length ARRAY_SIZE (you can declare that as a static final variable) then fill them with random values.
- First sort them one at a time, then refill them and sort them in N threads. Use System.nanoTime to time how long each of those takes.

Abstract Data Types (ADTs)

- Next class we will be working with the simplest forms of abstract data types. These are things that hold data and specify how you can interact with it and what happens when data is added or removed.
- In Java an ADT is basically an interface for a container with comments giving details on what happens with each method.
- Note that it doesn't specify how things happen. That is why it would be an interface. ADTs can be implemented in many different ways.



Stacks and Queues

- The simplest forms of ADTs, they each require one method to add an element and one method to remove an element. For easy of use we typically also include two other methods.
- Methods of a stack
  - push
  - pop
  - peek
  - isEmpty
- Methods of a queue
  - enqueue
  - dequeue
  - peek
  - isEmpty

The Difference?

- Push and enqueue add items while pop and dequeue remove items. The difference is what item gets removed.
- A stack is last in, first out (LIFO). Just think of how you interact with a stack.
- A queue is first in, first out (FIFO). If you were British you would use the term queue instead of line for what you stand in when waiting for something.
- Next class we will implement both of these using arrays.

Minute Essay

- Can you think of any programs that you would want to multithread? How might you break the work up in that program?
- Have you done any Java coding outside of class that isn't part of the assignment?
- Remember to turn in assignment #2 today.
- Quiz #2 will be at the beginning of next class.