

Slide 1

### Administrivia

- Reminder: Homework 2 design due today, code Tuesday.
- There are “practice problems” on the Web that you can try if you feel like you need them.

Slide 2

### More Administrivia

- Some of you may want to do some or all development using your own machines. You can do that, but you may want to also have a copy of your code on our file servers. Two options for transferring files between machines:
  - Simple, not particularly smart: Just copy `.java` files between machines. Several options.
  - More complicated to get started, more professional: Use CVS (version-control software, Eclipse has built-in support).
- Some instructions in project description.

## Homework 2, Again

- (Short demo of screen editor program.)

Slide 3

## Arrays in Java — Review

- Declaring and creating arrays in Java is different from in C — examples:  

```
int[] x = new int[10];  
String[] s = new String[n];
```
- Once created, though, some things are familiar — syntax for finding elements, range of indices.  
(Notice, though, that the second example above creates not `String` objects, but *references* to `String` objects.)
- Under the hood, more differences — in C, arrays are almost indistinguishable from pointers, but in Java, they're objects, with a `length` field you can use (but not change), and built-in bounds checking.
- Arrays as parameters to methods — what is passed is a *reference* to the array, so the method can change its elements.

Slide 4

## Sorting and Searching Arrays

Slide 5

- A common thing to do with arrays is sort them. (In theory this is covered in PAD I, but in practice, not always, so we will spend time on it.)
- Various algorithms for sorting and searching. Some fast, some slow; some simple, some complex. Decide which to use based on considerations of simplicity versus speed.
- "Speed"? Yes, but expressed as order of magnitude ("big-oh notation").

## Simple (but Slow) Sorts

Slide 6

- Bubble sort. (First pass goes through the whole array, swapping consecutive elements if out of order, so largest element bubbles to the end. Next pass goes through all elements but last. And so forth.)
- Selection sort. (First pass finds largest element and puts it at end. Next pass finds next-to-largest element and puts it at next-to-end. And so forth.)
- Insertion sort. (First pass inserts second element into list of first element. Next pass inserts third element into list of first two elements. And so forth.)
- (Code example.)

### Other Sorts

- Other comparison-based sorts (to be discussed later) include quicksort and mergesort.
- Other methods include bucket sort and radix sort.

Slide 7

### Searches

- Sequential search — start with the first element, examine elements one after another until a match is found or there are no more to examine.
- Binary search (for sorted data only) — examine the middle element and either stop if a match is found or recursively search the left or right half of the array.

Slide 8

## Minute Essay

- None — quiz.

Slide 9