Slide 1

Administrivia

- Reminder: Homework 2 due today.

Slide 2

Example Continued

- (Finish/review root-finding example from previous class.)

Slide 3

Arrays in C, Briefly

- Syntax for creating arrays is somewhat different from Java's — no explicit `new`, but instead something like

  ```
  int x[10];
  ```

  to reserve space for 10 `int`s. In old-style C, sizes must be constants known at compile time. In new-style C, "variable-length arrays" (VLAs) are permitted as well.
- Syntax for array access is the same as Java, but there's no `length` variable, and no checks are made to ensure that the index is legit (between 0 and array size minus one). This can make for interesting bugs . . .
- Syntax for passing arrays as parameters to functions is somewhat like Java's, except brackets typically go after the parameter name, and and arrays and pointers (more later) can be used more or less interchangeably.

Slide 4

Strings in C

- Java has a `String` class with many useful features and methods. In C that's not possible . . .
- Instead, in C, strings are arrays of `char`s, with the convention that the actual text of interest is followed by a null character (8-bit zero, represented in code as `'\0'`.
- You can operate on individual characters however you see fit; there are also standard library functions for some common operations (e.g., `strcmp` to compare two strings — similar to `compareTo` in Java).
- A significant source of potential trouble — most functions assume that strings are properly terminated, and (worse) many have no safety check to make sure you don't overflow a destination array.
- (Examples, etc., next time.)