Slide 1

Administrivia

- Homework 2 to be on Web later today or tomorrow. Due next Monday.

Slide 2

Variables

- Simple variables (numbers, characters, etc.) are fairly similar to Java primitive variables. Key differences:
  - Sizes of numeric types aren't as strictly defined — e.g., a Java `int` is exactly 32 bits, but a C `int` may be more. (Why? to allow implementations to use whatever is most efficient.)
  - No `boolean` in C89.
  - `char` is an ASCII (not Unicode) character.
- Arrays syntactically similar to Java, but more primitive (more about them later).
- Pointers — later.

Slide 3

Expressions, Statements, and Control Structures

- Most syntax is similar to Java (which is no accident) — within each function, code is organized into statements, which may contain expressions.
- Control structures are mostly the same as in Java — `if` , `while`, `do`, `switch`, `for`, etc. C doesn't have the simpler/newer form of `for` (referred to as "foreach").
- Key difference is the lack of classes (and supporting syntax), and addition of pointers (more later).

Slide 4

Functions

- Functions also are similar to those in Java, with a couple of key distinctions:
  - They have to be declared (or defined) before being referenced.
  - Pass-by-value semantics for parameters means you need pointers if you want to modify/return more than a single value.
- Library functions (e.g., `printf`) documented in `man` page. To use them, be sure to include the appropriate `#include`.

Slide 5

Sidebar — Compiler Options

- Earlier I showed the simplest way to use `gcc` to compile a program. But there are many variations — *options*. Specify on the command line, ahead of name of input file.
- Some of the most useful:
  - `-Wall` and `-pedantic` warn you about dangerous and non-standard things.
  - `-std=c99` allows you to use full C99.
  - `-o` allows you to name the output file (default `a.out`).

  (The right way to use all of these — makefiles, next time.)

Slide 6

Examples

- First let's write a program to calculate the roots of a quadratic equation, using the quadratic formula. (We'll hard-code input values for now — a discussion of getting input should wait until after we talk about pointers next time.)

Slide 7

Minute Essay

- There's a bug in the program we wrote in class; for some value(s) of the inputs, the function gives wrong answers or crashes. What input(s) give wrong results?

Slide 8

Minute Essay Answer

- If a is zero, the function tries to divide by zero, which will either crash or give a wrong result.