

## Administrivia

- Homework 2 on Web. Due next week.

Slide 1

## Variables in C

- 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 similar to Java references, but more flexible / less safe.

Slide 2

### Expressions, Statements, and Control Structures

Slide 3

- 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 use of pointers rather than references.

### Functions

Slide 4

- 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`.

### Sidebar — Compiler Options

Slide 5

- 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.)

### Examples

Slide 6

- 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.)

### 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 `ints`. 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 arrays and pointers (more soon) can be used more or less interchangeably.
- (To be continued.)

Slide 7

### Minute Essay

- None — sign in.

Slide 8