

Programming Basics (as described in CSCI 1320)

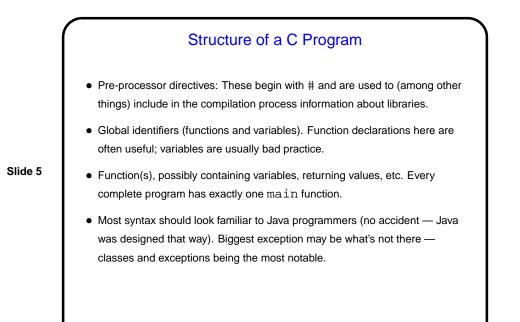
- What computers actually execute is *machine language* binary numbers each representing one primitive operation. Once upon a time, people programmed by writing machine language (!).
- Slide 2
- Nowadays, "programming" as we will use it means writing *source code* in a *high-level language*. Source code is simply plain text, which ... At this point we diverge from the explanation for beginners. Exactly what happens to get from source code to something the computer can execute varies among languages ...



- Some high-level languages (such as the language understood by typical UNIX/Linux command shells) are directly interpreted by some other program.
- Others are *compiled* into *object code* (machine language) and then *linked* with other object code (including system libraries) to form an *executable* (something the operating system can execute).
- Slide 3
- Java takes a somewhat intermediate approach it's initially compiled into *byte code* (object code for a made-up processor), which is (in principle) interpreted by the runtime system (Java Virtual Machine), with system library code brought in at runtime. (In practice, often a "just-in-time" compiler translates byte code into native object code on the fly.)

Why Learn C? (For Java Programmers)

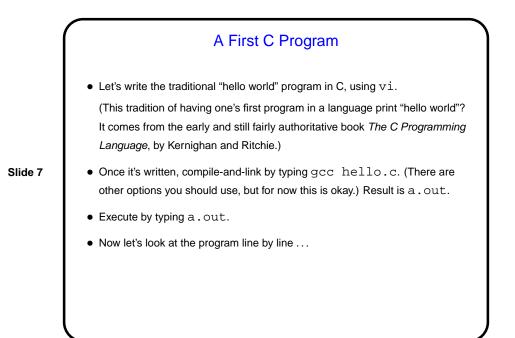
- Java provides a programming that's nice in many ways lots of safety checks, nice features, extensive standard library. But it hides a lot about how hardware actually works.
- Slide 4
- C, in contrast, has been called "high-level assembly language" so it seems primitive in some ways compared to Java. What you get (we think!) in return for the annoyances is more understanding of hardware — and if you do low-level work (e.g., operating systems, embedded systems), it may well be in C.

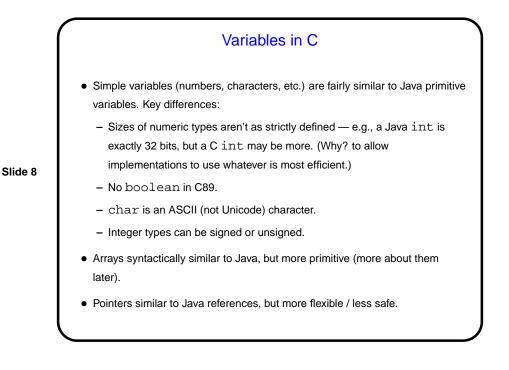


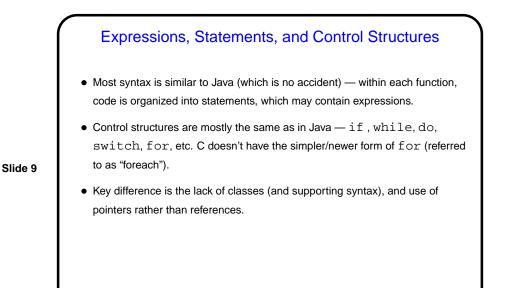
A Few Words About "Old C" Versus "New C"

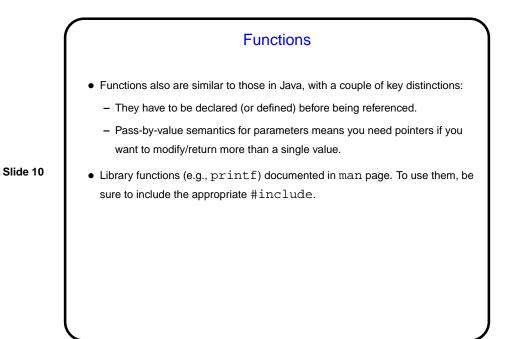
- First ANSI standard for C 1989. Widely adopted, but has some annoying limitations.
- Later standard 1999. Many features are widely implemented, but few compilers support the full standard, and older programs (and some programmers concerned about maximum portability) don't use new features. Much of what we do in this class will focus on older standard for this reason.

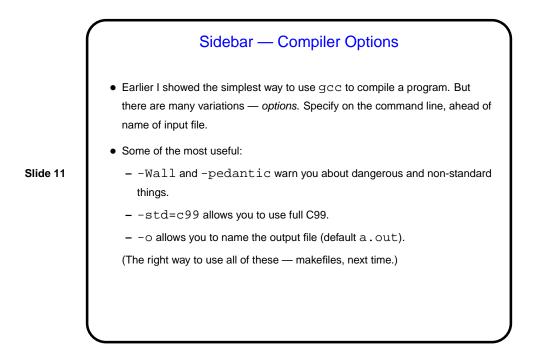
Slide 6

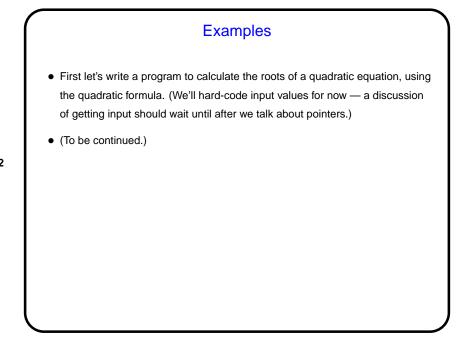












Slide 12