

"Hello World" Program Revisited

- Look again at the program we wrote in class previously. Most of it is standard boilerplate, to be discussed further soon. Single line you should pay attention to now is the one with printf.
- Goal for today describe how to extend this to get input from "standard input" (keyboard by default), do simple computing, write results to "standard output" (terminal window by default).
- (Maybe worth pointing out now that this is the style of programs we'll write simple text-mode user interface. This is really all you can do in standard C.)

Sidebar: C Design Goals

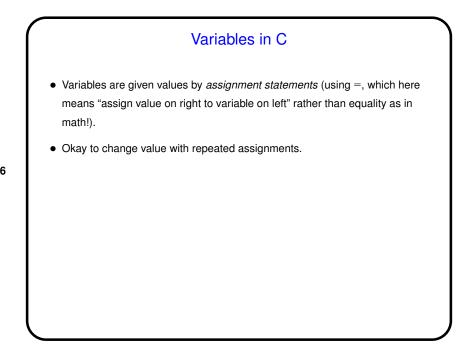
- Many currently-popular languages are big and complicated and designed to make the programmer's job easier. Often they include huge "libraries" to support interesting features such as GUIs, graphics, network communication, etc., etc.
- C, in contrast, was intended to be efficiently implementable on a very wide range of "platforms" (combination of hardware and operating system). It's therefore somewhat minimalist. GUIs, graphics, etc., can all be done from C, but only by using libraries that aren't part of standard C. And efficiency takes precedence over programmer convenience.

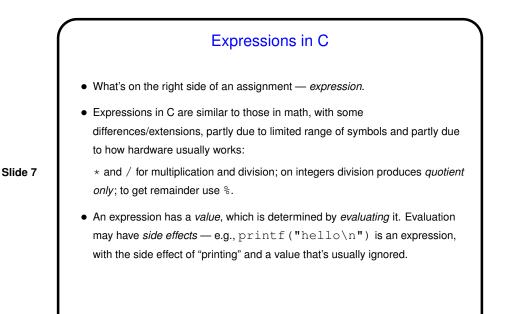
Slide 3

Slide 5



- storage for data e.g. someplace to save an input value and/or intermediate results. For this we use *variables*.
- In C variables must be *declared*, each with both a name and a *type*. Effect of declaring a variable is to reserve memory (RAM) for a value of the specified type and give it a name that can be referenced. (Similar to Matlab, except for choice of types?) What a name can look like is somewhat restricted (see textbook).
- Types in C are pretty basic integers, "floating-point numbers" (numbers with a fractional part), and characters. Integer types are represented as fixed-size binary numbers and come in various sizes. More about the others later.

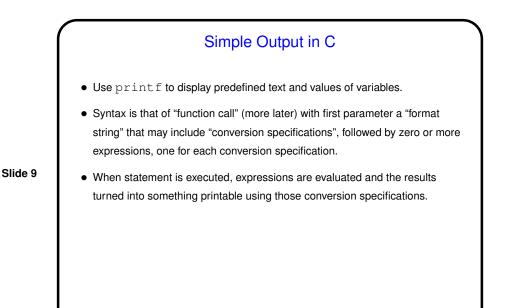




Assignment Statements Revisited

• Simplest programs are often basically a sequence of assignment statements (plus some "statements" that are really just expressions, such as that printf in the "hello world" program).

• Unless otherwise indicated, statements are executed in the order in which they appear in the code. (Sequential-ness is important and sometimes trips up beginners.)



Simple Output in C — Conversion Specifications
Conversion specifications say what kind of data is to be printed (integer, floating-point, etc.) and how.
For example, %d prints an integer in base 10, %x prints an integer in base 16. Also options to print with a fixed width (so output lines up in columns), control number of digits after the decimal point, etc.
man 3 printf for all the details. (There are a lot.) (What's that "3"? There are several things called printf; the 3 says we want the C library function.)

