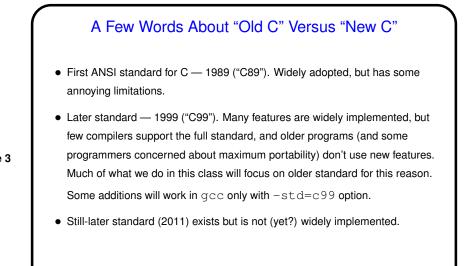
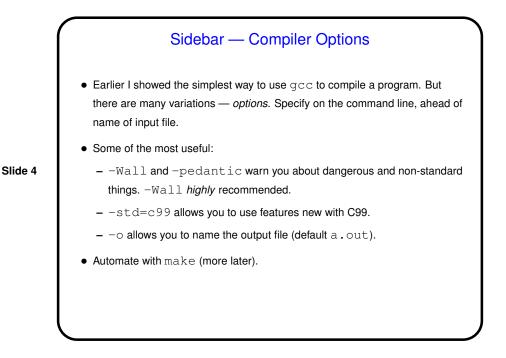
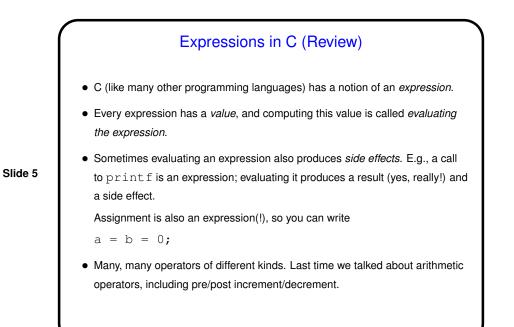


A Few More Words About vim

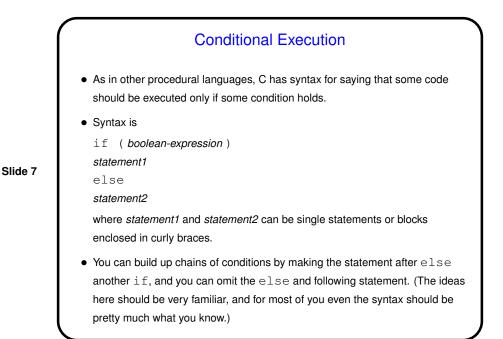
- Most people seem to have learned something from the tutorial. Good! vim is painful to use if you know only the bare minimum but starts to seem reasonable when you know more.
- I have an introduction to vim linked from the class "Useful links" page. The first part you may know, but at the end there's a section about behaviors that may have puzzled you.
- Two keys to know about if you don't already: u to "undo" previous command, and % to find matching parenthesis/brace. Also you might like "visual mode";
 :help visual-mode to learn more.

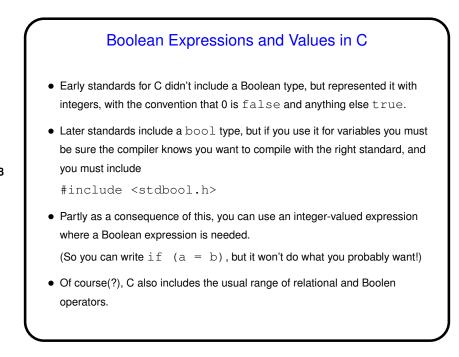


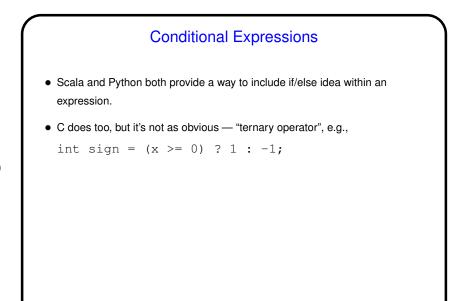


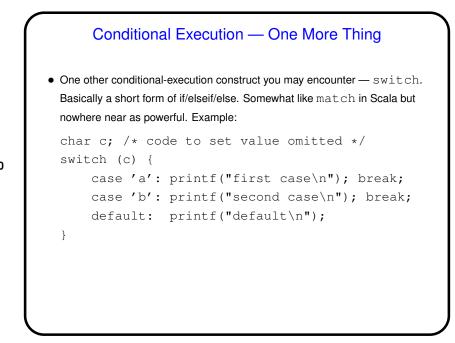


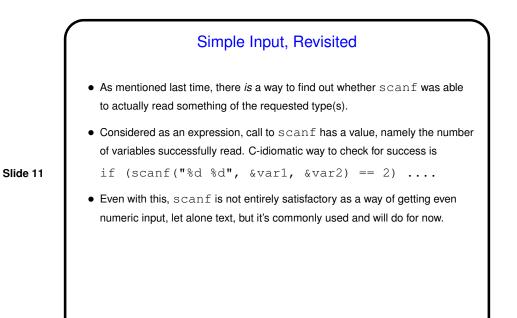
	Expressions — "Caveat Programmer"
•	Expressions can be quite complex. How they're evaluated depends on rules of precedence and associativity. My advice — when in doubt, use parentheses! Example: $(x + y) / 2$ versus $x + y / 2$.
•	C standard is somewhat imprecise about details of expression evaluation — e.g., in evaluating f () + g ()
	two functions could be called in either order. (Why? To allow greater flexibility for implementers, possible allow for more-efficient programs.)
•	C syntax allows programmers to write statements/expressions in which a variable's value is changed more than once, e.g., i = (i++) + (i); Syntactically legal, but standard says that such expressions invoke "undefined
	behavior". Best to avoid that!











Functions in C
Functions in C are conceptually much like functions in other procedural programming languages. (Methods in object-oriented languages are similar but have some extra capabilities.)
I.e., a function has a *name*, *parameters*, a *return type*, and a *body* (some code).
Slide 12
One difference between C and higher-level languages: You aren't supposed to use a function before you tell the compiler about it, either by giving its full *definition* or by giving a *declaration* that specifies its name, parameters, and return type. The function body can be later in the same file or in some other file.
Also, C functions are not supposed to be nested (though some compilers allow it).

