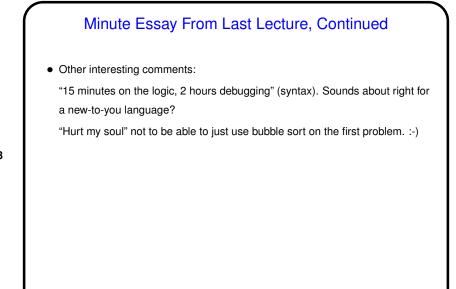


Minute Essay From Last Lecture

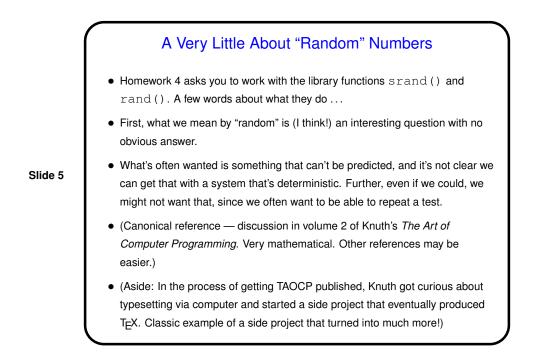
- About seconds per hour, etc., in Homework 2, no clear majority: Some used a calculator (or the equivalent — one mention of Python!), other a search engine.
- About Homework 3, several mentions of difficulty checking for non-integer input. Before searching the Web, consider reviewing examples on "sample programs" on course Web site — I mean for them to be useful in doing the assignments.
 - Also several comments about the first problem yes, it's just a specialized sort, and can be done that way (see one of my sample solutions), or with conditionals.
- Slide 2



Slide 3

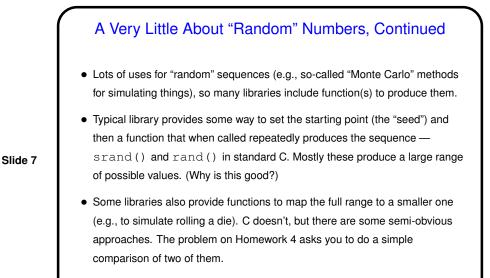
Slide 4

gcc Tip I say always always compile with -Wall — six extra keystrokes, and not even that if you remember about the up arrow in bash (shell). And then *do something* about warnings — almost all indicate a potential problem! (If you can't figure out what, ask! if nothing else asking me by e-mail works though isn't as immediate.) (The first thing I usually do when students ask why their code doesn't work is to ask them to recompile with this option. It's surprising — or not! — how often it warns about something that turns out to be the source of the problem!) If you want to be really thorough, add -pedantic (flags nonstandard usage — such as nested functions).



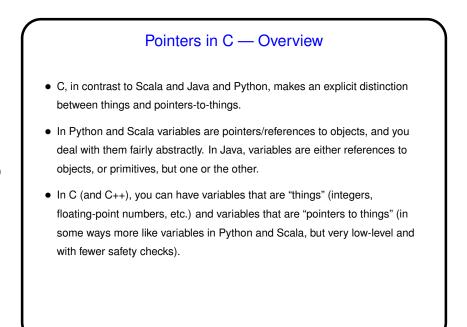
A Very Little About "Random" Numbers, Continued

- So, often what we really want is a "pseudo-random number generator" something that generates a sequence of numbers that looks random but is repeatable given some reproducible starting point.
- Slide 6
- Early researchers apparently thought more-complex algorithms would give better results, but — not necessarily. Very simple algorithms can give quite good results!



Arrays - Review/Recap • As in other languages, arrays give you a way to create the an indexed collection with all elements of the same type. • Unlike most modern(?) languages, arrays in C are a thin veneer over the implementation and lack safety checks and object-oriented features such as built-in length.

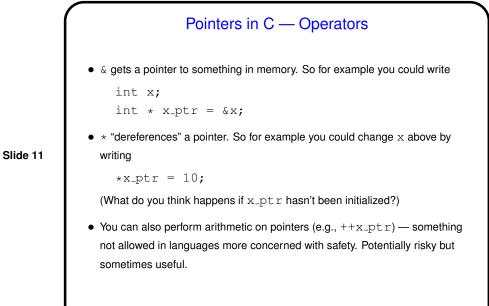
Slide 8

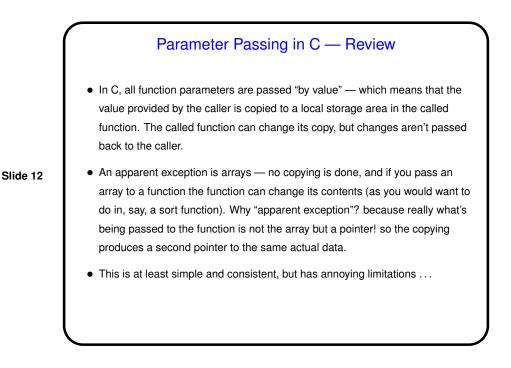


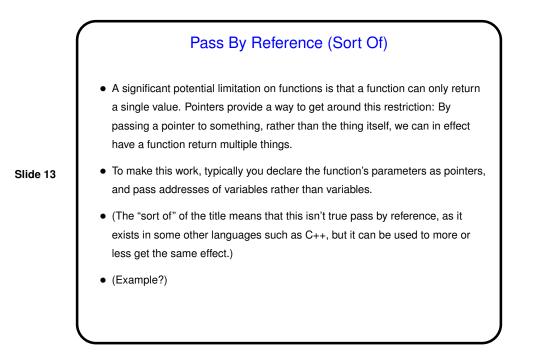
Slide 9

Slide 10

Pointers in C — Overview Continued That is, in C, pointers can be thought of as memory addresses (indices into large one-dimensional memory space — not always strictly true but a good first approximation), though declared to point to variables (or data) of a particular type. Example types: int * pointer_to_int; double * pointer_to_double;

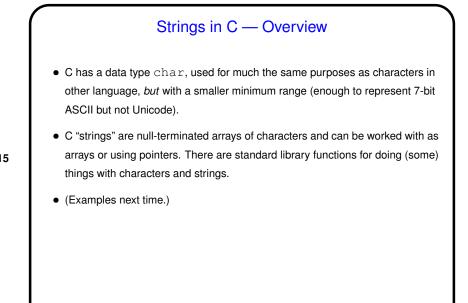






Pointers Versus Arrays In almost all contexts arrays and pointers are interchangeable. In particular, if you declare the type of a function parameter to be a pointer, you can pass it an array, and vice versa.

Slide 14



Minute Essay
Do you remember to compile with -Wall? and if so, do you try to fix anything being warned about? I ask because I got a fair amount of code for Homework 3 that gave warnings ...
Any questions — about pointers, strings, anything else?

Slide 15