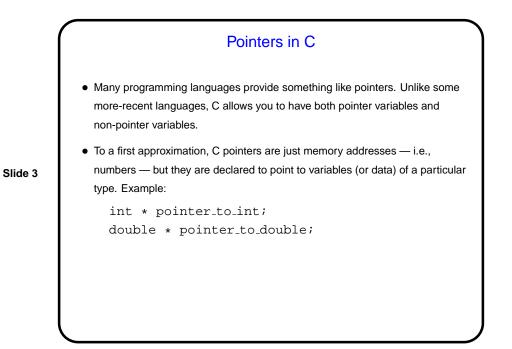
Slide 1

## Administrivia

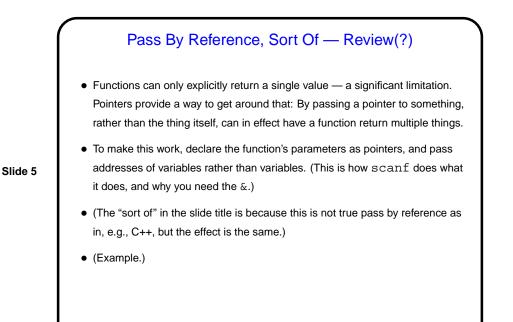
- Reminder: Quiz 4 Monday. (If you cannot reasonably be present then, talk to me about whether it's possible to make up this quiz later.) Likely topics include material from this week and the little we did about sorting and "order of magnitude" of algorithms. (So, review how the three sorting algorithms in the textbook work, focusing on how they move data around rather than on code.)
- Next homework still "in work" ...

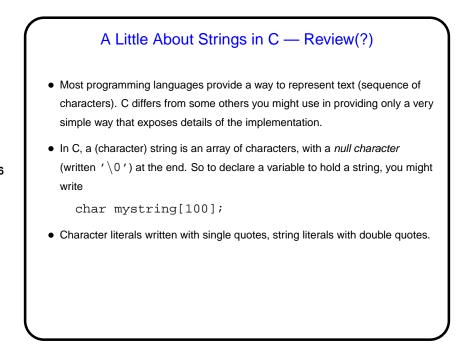
## **Pointers Revisited**

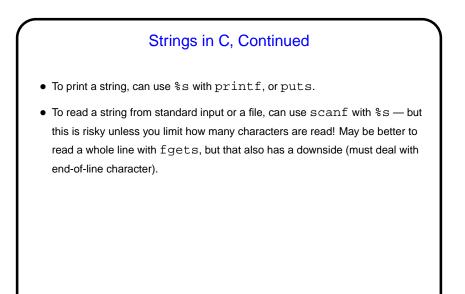
- Every time you call scanf, you pass it at least one parameter of the form &x. What does that mean? Also, when you look at man pages for some functions, they show function declarations with parameters of the form *type* \*. What does that mean?
- Slide 2
- To explain, we need one more kind of variable *pointers*. A pointer, as its name suggests, points to something namely, a location in memory.
   Typically a pointer "points to" a variable.



Pointers in C — Operators
• & gets a pointer to something in memory. So for example you could write
 int x;
 int \* x\_ptr = &x;
• \* "dereferences" a pointer. So for example you could change x above by
 writing
 \*x\_ptr = 10;





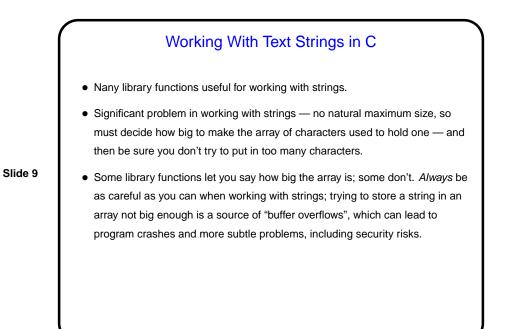


Slide 7

## Pointers, Arrays, and Pointer Arithmetic in C

- C treats pointers and arrays as interchangeable in most respects. (This is why it works that many functions whose parameters are supposed to be strings arrays of characters declare them as pointers. Many many examples ...)
- C also permits doing some arithmetic operations on pointers (addition and subtraction). Adding *n* to a pointer that points to *type* advances it *n* times the size of *type*.

Example: If a is an array of ints, a[2] and \*(a+2) are equivalent. (So we could write loops over arrays using pointers. Once upon a time that was sometimes more efficient. With current compilers, probably not so, so use whatever is most readable.)



Working With Text Strings in C, Continued

Many library functions for working with strings use/return pointers. Pointer arithmetic allows for some interesting uses of these functions.

• (Examples as time permits.)

