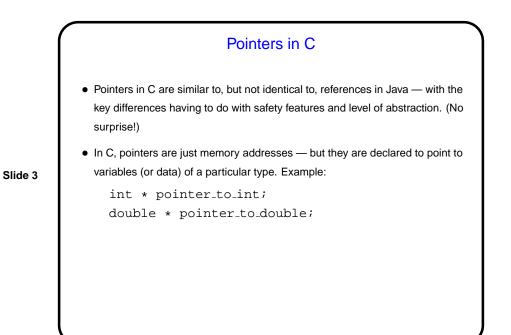


Slide 2



Pointers in C — Operators
& gets the address of 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;

You can also perform arithmetic on pointers (e.g., ++x\_ptr) — something not allowed in Java, and another example of the languages' different design goals.

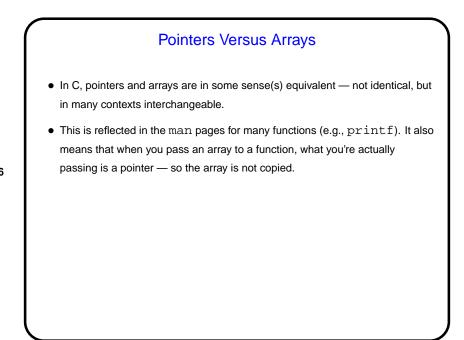
Slide 4

## Pass By Reference (Sort Of)

• A significant potential limitation on functions is that a function can only return a single value. Pointers provide a way to get around this restriction: By passing a pointer to something, rather than the thing itself, we can in effect have a function return multiple things.

Slide 5

- To make this work, typically you declare the function's parameters as pointers, and pass addresses of variables rather than variables.
- The "sort of" of the title means that this isn't true pass by reference, as it exists in some other languages such as C++, but it can be used to more or less get the same effect. Notice also that Java can't do this, though again there are mechanisms that can more or less get the same effect. (What?)



Slide 6

