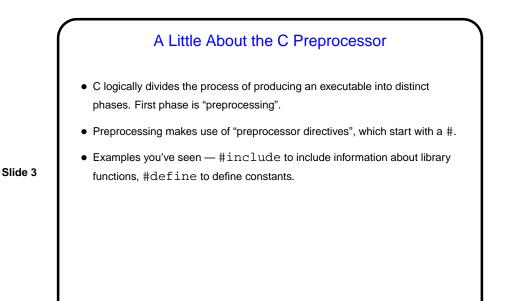
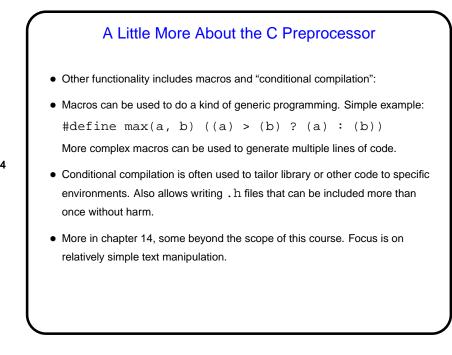


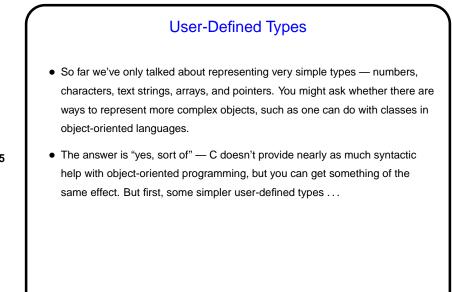
## Minute Essay From Last Lecture Many people sort of got the key point (see "answer" slide) — if you let "the system" manage memory, it's easier and less error-prone, but you have less

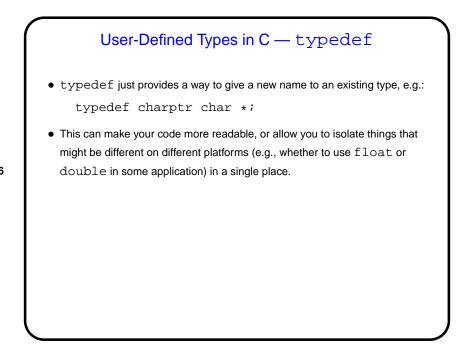
Slide 2

control.











• In C (and in some other programming languages) an *enumeration* or an *enumerated type* is just a way of specifying a small range of values, e.g.

```
enum basic_color { red, green, blue, yellow };
enum basic_color color = red;
```

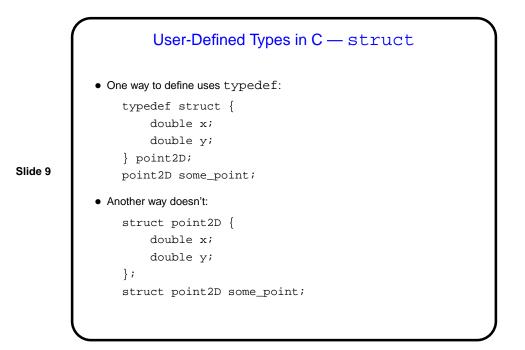
- This can make code more readable, and sometimes combines nicely with switch constructs.
- Under the hood, C enumerated types are really just integers, though, and they can be ugly to work with in some ways (e.g., no nice way to do I/O with them).

## User-Defined Types in C — struct

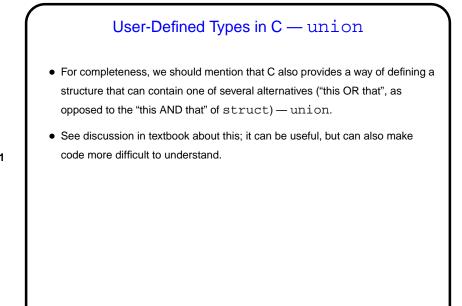
 More complex (interesting?) types can be defined with struct, which lets you define a new type as a collection of other types — something like a class in an object-oriented language, but with no methods and no way to hide fields/variables.

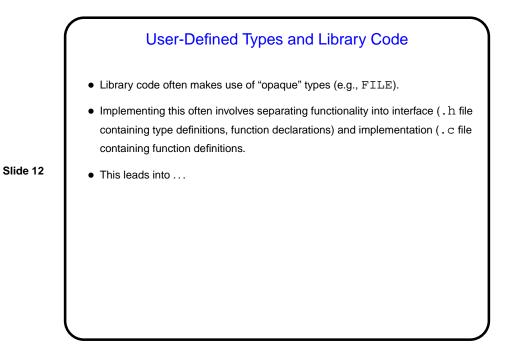
Slide 8

• Two versions of syntax (next slide) ...



# User-Defined Types in C — struct, Continued • Either way you define a struct, how you access its fields is the same: if what you have is a struct itself: struct point2D some\_point; some\_point.x = 10.1; some\_point.y = 20.1; -> if what you have is a pointer to a struct: struct point2D \* some\_point\_ptr = &some\_point; some\_point\_ptr->x = 10.1; some\_point\_ptr->y = 20.1;





## Separate Compilation — Review

 C (like many languages) lets you split large programs into multiple source-code files. Typical to put function declarations (headers), constants, etc., in file ending . h, function definitions (code) in file ending . c.
 Compilation process can be separated into "compile" (convert source to object code) and "link" (combine object and library code to make executable) steps.

Slide 13

 UNIX utility make can help manage compilation process. Can also be useful as a convenient way to always compile with preferred options. Idea behind make — have computer figure out what needs to be recompiled and issue right commands to recompile it.

### **Makefiles**

• First step in using make is to set up "makefile" with "rules" describing how files that make up your program (source, object, executable, etc.) depend on each other and how to update the ones that are generated from others. Normally call this file Makefile or makefile.

Slide 14

• When you type make, make figures out (based on files' timestamps) which files need to be recreated and how to recreate them.

Simple example on sample programs page.

