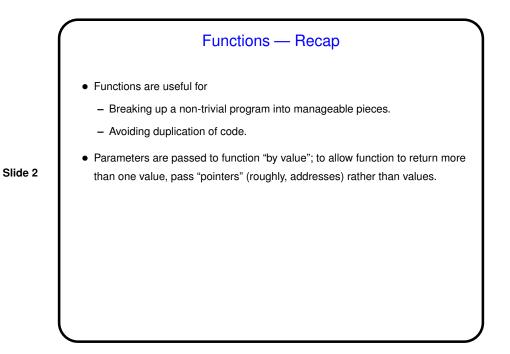
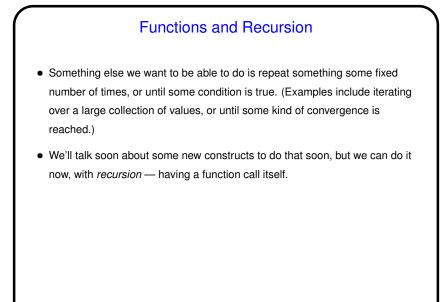
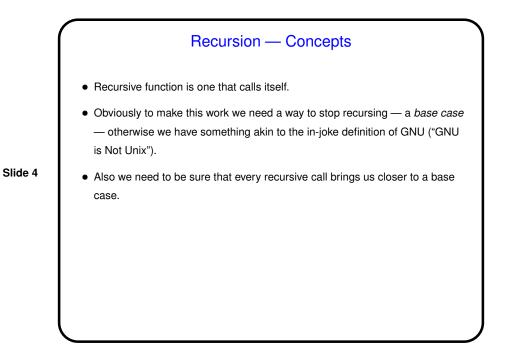


Slide 1





Slide 3



Recursion — Implementation

• How it works: When you call any function, the current "state" (values of variables) is preserved ("pushed onto a stack"), and space is reserved for the called function's local variables (including parameters). When the function returns, this space is freed up again. So if we stack up recursive calls to the same function, each has its own copy of all local variables.

Slide 5

• Simple examples — factorial, Fibonacci numbers, counting(?), sum from input.

Minute Essay

• Many people did not get full credit for the first question on the quiz. I didn't spend a lot of class time on this material, thinking a lot of it was review for many of you. Maybe not? It's far from the most important topic in the course, but I'd like most of you to get it. Review more in class Friday or soon?

• Anything noteworthy to say about Homework 3?

Slide 6