Slide 1

Administrivia

- Homework 4 on the Web. Due next Thursday.

Slide 2

Minute Essay From Last Lecture

- Question was about frequently-made mistakes.
- Some answers:
  - Making sure parentheses and curly brackets match up. (Remember `vi` tip.)
  - Syntax details in general.
  - That `a < b < c` doesn't do what you might want.
  - That you have to save the program code before trying it again. Also problems with being in the wrong directory or forgetting the name.
  - Modes in `vi`.

Miscellaneous Tips

- Remember that you can save some typing using the up/down arrow keys to cycle back through previously-entered commands. Also, typing part of a filename and pressing tab will fill in the rest (if there's only one that matches).
- If you start up your program and it prompts for something but you just want to quit, you can type control-C and it will be interrupted. (Many UNIX command-line tools use control-C this way.)
- When you click on a filename in the graphical file manager it's opened using some application (which one depends on filename). For `.scala` files the application appears to be `emacs`, which is another text editor. You may be better off opening a terminal window and using `vi`, if you also might want to make changes.

Ways to Use Scala — Review

- One way to use Scala is interactively — `scala` with no filenames. Very useful for experimenting with what various things do, and the interpreter keeps a history (which you can cycle back through with up/down).

  When in this mode you can use `:load` to load files containing functions. Can be useful as a quick way to test functions, but does require that definitions of functions come before their use.
- Another way is as a scripting language — `scala pgm.scala`. Executes the statements in `pgm.scala`. Notice that in this mode definitions of functions do *not* need to come before their use.

Slide 5

Sidebar: Input/Output Redirection

- Normally programs run from the command line write output to the terminal window. Can instead "redirect" output to a file:

  `> outfile` (overwrite)

  `>> outfile` (append)

- Normally programs get input from the keyboard, but can also make them get input from a file with <.

  (How could this help you in checking your programs?)

- Finally, can use "pipes" (vertical-bar `|`) to have output from one program become input to another. Example:

  `ruptime | grep xena` (show status of HAS 340 machines)

  Very powerful idea! this and some other ways of connecting simple programs makes for a very powerful and flexible environment.

Slide 6

Repetition and Recursion — Overview

- Having if/else allows us to do a lot of things we couldn't do before, but there are still things we can't do easily, mostly involving some sort of repetition. Simple example — adding something to the grade program that would prompt for six quiz scores. Another example might be trying to use our bounding-box function to find a bounding box to enclose more than two rectangles, with the choice of how many up to the user.
- Scala provides many ways to do this. We will look at recursion first.

Slide 7

Recursion

- Basic idea of recursion is to solve a problem by defining
  - “base cases” we can easily, and
  - a way of reducing other cases to “smaller” instances of the problem
- Simple examples abound in math; a traditional first example is computing the factorial of an integer. We can define $n!$ as the product of the integers from 1 through n, or we can use a recursive definition:

$$n! = \begin{cases} n \cdot (n-1)! & \mathit{if}\ n > 1 \\ 1 & \mathit{otherwise} \end{cases}$$

  This is easy to convert into code in a language that supports recursion . . .

Slide 8

Recursion, Continued

- Key ideas in recursion:
  - One or more base cases that can be solved without recursion.
  - A way of splitting up other cases into one or more *smaller* recursive calls plus some other logic.
- Very important that recursive calls be somehow smaller, so that you eventually reach a base case!
- As one more example for now — function to “count down” (print numbers from starting point through 1).

Minute Essay

- None — quiz.