

Slide 1

Administrivia

- Quiz 1 sample solution available online. Grades rather disappointing. However, only 10 points, and I do drop the lowest quiz score.
- Reminder: Homework 3 due today.
- Homework 4 on the Web. Due next Wednesday.

Slide 2

Functions Example, Revisited

- We wrote a program to find and print the roots of a quadratic equation.
- The basic calculation is one that you might want to use in another program. So to me it makes sense to “package” it as a function, so it would be easy to copy to another program. (Can also put it in a separate file. Later!)
- I like to also write a main program that incorporates some test cases. For that I often define a secondary function that prints input and output of a test case, as in latest revision of the quadratic-equation program.
- Note also that latest revision uses `switch` to go through various cases in function to print.

Functions and Recursion

Slide 3

- Something else we want to be able to do is repeat something some fixed number of times, or until some condition is true. (Examples include iterating over a large collection of values, or until some kind of convergence is reached.)
- We'll talk soon about some new constructs to do that soon, but we can do it now, with *recursion* — having a function call itself.

Recursion — Concepts

Slide 4

- Recursive function is one that calls itself.
- Obviously to make this work we need a way to stop recursing — a *base case* — otherwise we have something akin to the in-joke definition of GNU (“GNU is Not Unix”).
- Also we need to be sure that every recursive call brings us closer to a base case.

Recursion — Implementation

Slide 5

- 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.
- Simple examples: factorial, Fibonacci numbers, sum from input.

Minute Essay

Slide 6

- Have you encountered recursion previously? perhaps in a math class? how about proofs by induction (sort of a similar idea)?
- How did Quiz 1 compare to your expectations (length, topics, difficulty, etc.)? If you lost points, what do you think went wrong?