2-6-2012
Opening Discussion

- Minute essay comments
  - Can Scala do complex math? \((z=a+bi)\)
  - Graphing functions.
  - 3-D functions.
  - Limit on number of parameters?
  - Derivatives of functions? Not all functions are mathematical.
- Test questions compared to IcPs.
- Read error messages.
- Functions can nest.
I welcome our new robot overloads.

When programming for a living, how much is math and how much is logic and words?

Placement of functions and sequential execution.

Multiple source files.

Pitfalls to relying on functions?

Applications for ray tracing.

Local variables.
We declare functions in Scala using def. Here is the general form.

- `def name(arg1: Type1, arg2: Type2, ...): Type = expression`

The argument list can have zero or more elements. If there are zero even the parentheses can be left off.

- Function arguments must have types.
- The return type is optional, but it is recommended.
Why Functions?

- Functions are used in programs for a number of reasons.
  - Reduce code duplication. You can call the same function multiple times and only write it once.
  - Improve readability and maintainability. Good function names make it easier to read. Small functions are easier to test and debug.
  - Break problems down/problem decomposition.
Problem Decomposition

- Never solve a hard problem. If a problem is hard, break it into smaller problems that are easier. Repeat until you are only solving trivial problems.

- Top-down
  - This is the “normal” approach where you start with the full problem and break it into pieces.

- Bottom-up
  - Sometimes you realize that different trivial pieces will be useful and build up from those.
Function Literals

- Just like 5 is a literal for an Int and “hi” is a literal for a string, you can write literals of functions.

- The full syntax is an argument list followed by an equals arrow followed by the function expression.
  - (a:Int,b:Int) => 3*a+2*b

- Types don't have to be specified in many situations, only if Scala can't figure it out.
Higher-Order Functions

- These are functions that take functions as arguments or return functions.
- These are the main things we use function literals for. We will see them a lot in two weeks.
Assume I asked you to do some processing on 10000 numbers. How would that make you feel and why would you feel that way?