

Functions

9-22-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?
- Printing a number in binary.

Text Input in C: scanf

- To get input from the user with the stdio library you typically use the scanf function. It has a format very similar to that for printf where it begins with a format string. You use the same format signifiers with two exceptions.
 - %ld of longs and %lf for doubles
- Follow it with the addresses of variables you want to fill. (Put & in front of the variable name.)

Hex Numbers and Character Literals

- You can put numbers into your code in hex by preceding the hex value with "0x" so 0xFF is 255.
- You can do octal by having a leading 0.
- We saw last time that a string literal is denoted by surrounding it with double quotes. A character literal is denoted by putting single quotes around it.

Breaking Problems into Pieces

- A key to solving any complex problem is to break it into pieces that are of a more manageable size and solve each of those, then bring the elements together. This is like the outline view of an algorithm.
 - Top-Down Design: Starting with the whole problem, repeatedly break it into pieces going down to what you can handle.
 - Bottom-Up Design: Starting with little pieces work up to the full problem.

Functions in C

- The way we break problem up in C is to break our code into separate functions.
- You have already seen how we can call functions in our code. Now we look at how to write them.
- The first part of any function is it's signature:
`return-type name(type a1, type a2, ...);`
- C uses a "single pass" compiler so it must know about methods before they are called. Either put a signature, or the whole function at top.

Parts of a Signature

- C functions are much like normal math functions, but type matters.
- Return type: This is the type of what the function returns. It will be void if the function doesn't return anything.
- Arguments: This is a list of types and names for what is passed into the function. Again, if nothing is passed in it is void.

Function Body

- A method can be declared with a signature followed by a `;`. The function definition will have the signature followed by curly braces enclosing the statements that should be executed.
- As with main, you can put any type of C statements you want in there.

The return Statement

- When the statement "return" is reached, execution returns from the function to the point it was called from.
- For a void function you can use return; in the middle and don't need any at the end.
- For a non-void function, return should be followed by an expression of the correct type for what should be returned.

Formal vs. Actual Arguments

- Inside a function arguments act like variables. You don't know what values they will have in the function itself. Those are the formal arguments.
- At the point where the function is called, values are "passed" to the function. These are the actual arguments and determine what the formal arguments will be initialized to.

Minute Essay

- Write a function called minI that takes two integer arguments and returns the smaller of the two.
- Assignment #2 is due today and quiz #2 will be at the beginning of next class.
- Please read chapter 10 before next class.
