

Intro to C Programming

9-14-2002

Opening Discussion

- What did we talk about last class?
- Assignment #2 has been posted. I made a change Sunday afternoon so if you saw it before then, you need to look at it again.

Minute Essay Comments

- Role of binary/hex in programming. Also just part of general education in CS. We won't be doing this a whole lot during the semester.
- A bit more on fractional binary and floating point numbers.
 - Given a decimal fraction you find the binary representation the same way as for whole numbers.
- Replacing some instances in vi.
- Dual processor systems (fetch and execute). How the hardware in a computer relates to what we have talked about.

More Minute Essays

- C does use decimal numbers so you don't typically need to go to binary or hex. However, some techniques require you to understand them.
- C is a subset of C++ but the way you program them should be very different. I feel C is better for teaching at the intro level. What's up with the "<<" being a bit shifter?
- Negative numbers in the computer.
- We aren't going to talk about multiplying in anything beyond binary.

A First Example

- We will now look at some code that I have started to write and add a bit more to it.

Compiling and Executing

- Computers don't understand C code, they only understand machine language. For this reason we have compilers. They are programs that convert high level languages to machine code.
- The compiler you will use in this class is gcc. A man on gcc will give you the many options possible. Here are some:
 - -o: Executable output name. a.out is used by default.
 - -g: Include debug information.
 - -Wall: Print all warnings.
 - -pedantic: Only accepts tighter code.

Parts of a C Program

- #include lines: These lines tell the compiler what libraries you will be using and basically paste their code at the top.
- main: All C programs have a main and this is where the execution of the program begins. Main is a function. We will talk more about functions later. The body is inside curly brackets: '{' and '}'.
- Comments: All text between a /* and a */ is comments and are ignored by the compiler. They can span lines.

Statements

- The body of the main function consists of a series of statements that are executed in order, top-down and left-right.
- All statements end with a ';'. What comes before the ';' must be a valid C expression.
- Note that the statements in main are indented beyond the declaration of main. C ignores extra white space in the program.

Expressions

- An expression can be one of the following:
 - A number
 - A string literal
 - A variable name
 - A function call
 - Two expressions separated by an operator. For this, C must be able to perform the operation on the types of the expressions. Can include parentheses.
- An "atomic" element is called a token. The first three are tokens and adding white space changes their meaning.

Types and Variable Declarations

- C is a typed language so all expressions in C have a type. C has the following types:
 - char, short, int, long. These all represent integers and can be signed or unsigned.
 - float, double, long double. These represent floating point numbers.
- When you want to keep track of a value in C you declare a variable of the correct type. A variable declaration has the form of "type name1;".

Operators

- Complex expressions in C are built with operators. Here are the numeric operators available that take two arguments.
 - +, -, *, /: Do what you would expect.
 - %: Modulo, the remainder after division.
 - <<, >>: Bit shifting operators.
 - &, |, ^: Bitwise and, or, and xor.
- Here are operators that take one argument.
 - -: Negative.
 - ~: Bitwise negation.
- Tertiary Operator, ?:, takes 3 arguments.

Assignment

- There is also an operator '=' that is an assignment operator. It stores the value of the expression on the right hand side into the memory for what is on the left hand side.
- For the time being the only thing that will ever appear on the left hand side is a variable.
- You can do this in a variable declaration.

Functions

- The last somewhat atomic type of expression listed was a call to a function. You can use functions that exist in other libraries, like `printf` in `stdio`, right now. A bit later, we will learn how to define our own functions to help break up problems into smaller pieces.
- Function calls give the name of the function followed by an argument list in parenthesis.

printf the Short Version

- The `printf` function is how you will print things to screen. It allows you to do formatted output.
- For now you can think of its argument list as a format string followed by values to be put into that string at the `%?` tokens.
 - `%d` is for decimal integer
 - `%f` is for float
 - `%c` is for character
 - `%X` prints an integer in hex

Minute Essay

- Write a short C program that declares one integer variable (you pick the name) and stores the value of $2+2$ in it.
- Read chapter 3 for next class. The second half of the assignment could prove very challenging if your only information comes from lecture. Come to the next class armed with questions.
