

## **Problem Solving and Formatted I/O**

**9-19-2003**

---

---

---

---

---

---

---

---

## **Opening Discussion**

- What did we talk about last class?
- Do you have questions on assignment 2?
- Your file names for your programs should end with ".c". I have been accidentally using ".cpp" because I'm also doing a fair bit of C++ programming this semester.

---

---

---

---

---

---

---

---

## **What is an Algorithm?**

- Last time we looked at what the basics of a C program include, but we are also concerned more generally with how we get computers to solve problems.
- A set of explicit instructions for solving a problem is generally called an algorithm.
- Algorithms can have varying levels of detail, much like an outline. How much detail you need depends on who/what is going to be performing the actions.

---

---

---

---

---

---

---

---

## Blowing up a Balloon

- A standard example of an algorithm is making a peanut and butter sandwich. That's a bit messy for the classroom though. Instead, I would like for you to write an algorithm for blowing up a balloon. You will need to be fairly detailed because I'm going to be the one trying to perform the actions and I'm not that bright.

---

---

---

---

---

---

---

---

## 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.

---

---

---

---

---

---

---

---

## Text Output in C: printf

- The printf function is how you will print things to screen. It allows you to do formatted output.
- The printf function requires one argument that is a string. This string can contain special formatting characters that tell it to insert strings for variables or other values.
- There are also special "escape sequences" that allow you to print other characters.

---

---

---

---

---

---

---

---

## Formatting Characters

- The main formatting of the printf is done by putting in character sequences beginning with a '%' that say a later argument should be printed there.
  - %d is for decimal integer
  - %f is for float
  - %e does scientific notation
  - %c is for character
  - %X prints an integer in hex

---

---

---

---

---

---

---

---

## Escape Characters

- There are some characters that you might want to print that you can't easily put into a string literal. For these you use two character sequences that begin with a '\\'. These aren't technically related to printf, but are more general for all C strings.
  - \n is a new line character
  - \t is a tab
  - \a "prints" a beep
  - \b is a backspace

---

---

---

---

---

---

---

---

## 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.

---

---

---

---

---

---

---

## Code

- Now we will have you write some code that does some simple printing.

---

---

---

---

---

---

---

## Minute Essay

- From the last two lectures you have enough information to figure out how to print the binary representation of a number. Most of it deals with the operators we have looked at for integers. Describe how you might do this, or write code to do it for a few bits.
- Your book has a lot more information on detailed formatting of output that you might want to read.
- Assignment #2 is due on Monday.

---

---

---

---

---

---

---