

Functions and Parameters

9-14-2001

Opening Discussion

- What did we discuss in the last class?
- a.out is the executable created by default when you compile.
- Today we will move to the next major topic of functions. Before we leave numerical expressions, do you have any questions on that topic?
- What is a function and what does it allow us to do?

Math Library

- For this library with g++ you need the .h in the include statement. However, you don't have to link anything in.
- By putting `#include<math.h>` at the top of the program you will be able to use a number of math functions including pow, sqrt, log, fabs, sin, cos, tan, etc.

Errors

- Syntax errors - These are the easy errors because the compiler finds them for you and tells you about them. The way it tells you can be cryptic at times though.
- Runtime errors - This is where the code compiles properly but when it executes it crashes. Something like divide by zero.
- Logic errors - These are the real difficult ones to locate. Often originate in design.

Functional Decomposition

- Think back to our discussion of algorithms and the example of blowing up a balloon. Remember how I had you write things out in an outline format with more detailed instructions as subitems under the course instructions? What you were doing there was basically functional decomposition.
- It is often helpful to break a problem into pieces that can be solved independently. This is especially true when certain pieces come up multiple times.

Function Declarations in C++

- A function in C++ is similar to a function in math. It takes a number of inputs and returns a values.
- The format of a function declaration (also called a header or prototype) is

```
returnType functionName(arg1Type arg1Name,  
    arg2Type arg2Name, ...)
```

- The return types and argument types can be any valid types for the language (int, double, string, and many others). The return type can also be void if it returns nothing.

Function Definition

- You have already seen a function definition for main. After line giving return type, name, and arguments and open curly brace starts the function and it goes until a closed curly brace stops it.
- If the function has a non-void return type it must return a value. The syntax for this is:

```
return expression;
```

Formal and Actual Arguments

- The arguments listed in the declaration of the function are referred to as formal arguments. What is passed to the function when it is called are called the actual arguments.
- The two must agree in type or a conversion must occur for the call to actually compile.
- The names of the variables don't matter.

Pass by Value

- There are two ways to pass a value to a function. The most straightforward is pass by value.
- When an argument is passed by value a separate copy is made for use in the function. This means that changes to the variable made in the function don't alter the value of the variable in the calling routine.

Pass by Reference

- There are times when you will want a function to be able to change multiple values. This can be done by passing some arguments by reference.
- To see how this works think of the memory of the computer again as a series of boxes that can hold different value. When you pass by reference you tell the function where the box for that variable is. In contrast, pass by value sets up a new box and copies the value into that box.

Minute Essay

- Write a function declaration for a min function that returns the smaller of two integers.
- Remember that assignment #1 is due on Monday. In class I want you to hand in an analysis (English description of the problem you are solving including your assumptions), a design (English description of how you are going to solve that problem on the computer), and a print out of the code. You should e-mail the code to mlewis@bianca.cs.trinity.edu as described on the Linux help sheet.
