## Errors, Using Class Libraries, Etc.

**9-19-2001**

## Opening Discussion

- What did we talk about last class?
- Calculations before the values are set. The equal sign in C++ is not a a method of defining equality like in math. Instead it is a command to copy the value of one expression into the memory location of a specific variable. It uses the value of the expression at the instant it is executed.

## Reference Arguments

- The bottom line with a reference argument is that changes made to a reference argument have an effect on the value of the variable passed in from the calling function.
- Another way of saying this is that the changes to the formal arguments have an effect on the actual arguments.

## Variable Scoping (the part that really matters)

- The scope of a variable is just the region where it is usable in a program.
- A variable is usable between the line it is declared on and the end of the block it is declared in.
- The formal arguments of a function act as local variables with a scope of the entire function.
- Globals are declared outside of all functions and can be used anywhere.

## Global Variables

- The fact that globals can be accessed and used anywhere is dangerous because it means that they can be changed in any function. This is unless they are const in which case they are safe.
- If a value is needed in multiple functions it is probably safer as an argument. If it needs "memory" it is safer as a static local.
- We will cover other alternatives to globals later in the class.

## Static Variables

- A static local has the "memory" abilities of a global but the scoping of a local.
- Only initialized once. This can be deceptive because it looks like the initialization assignment would execute every time that line is passed in executing the program.
- This only matters is a function can be called multiple times. It is of little use in main.

## Return Value and return Operator

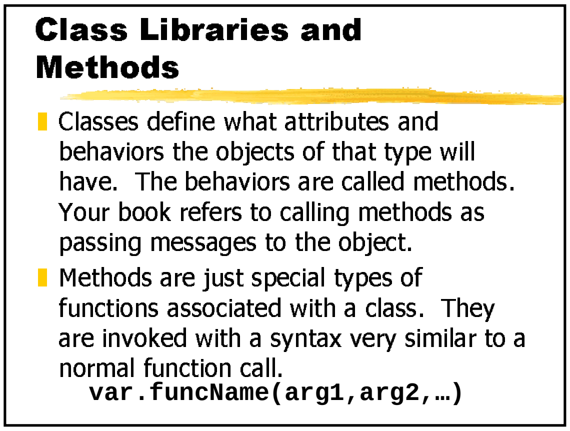
- Returning a value
  - When a function has a return type it must end with a statement "return *expression*" where expression has the type of the return type.
- Returning before the end of the function
  - You can also put a return statement before the end of a function at any place where the execution of the function should end and control should go back to the caller.

## Potential Errors

- In section 3.4 the book lists a number of syntax and usage errors that are likely to arise when you are writing functions in a program. It also includes some hints on style. Many of these things have come up already in our discussions. Some haven't.
  - No parentheses on a void function call.
  - Actual reference arguments should be variables.

## Objects and Classes

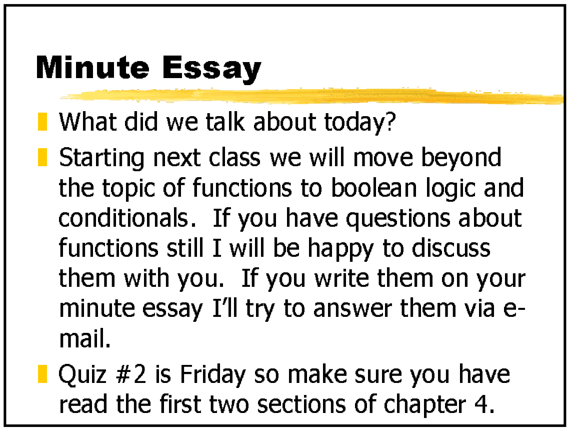
- C++ is a class based object-oriented language. Every object in a C++ program is an instance of a particular class. You might also hear of a class being instantiated to create an object.
- Later in the semester we will discuss how you create classes of your own, but even now you can use those created by others. For example cin and cout are instances of classes in the iostream library.



## Class Libraries and Methods

- Classes define what attributes and behaviors the objects of that type will have. The behaviors are called methods. Your book refers to calling methods as passing messages to the object.
- Methods are just special types of functions associated with a class. They are invoked with a syntax very similar to a normal function call.

```
  var.funcName(arg1,arg2,…)
```



## Minute Essay

- What did we talk about today?
- Starting next class we will move beyond the topic of functions to boolean logic and conditionals. If you have questions about functions still I will be happy to discuss them with you. If you write them on your minute essay I'll try to answer them via e-mail.
- Quiz #2 is Friday so make sure you have read the first two sections of chapter 4.