OOP Classes

CHAPTER 11

Experience is a dear teacher, but fools will learn at no other.

Benjamin Franklin

Coming up with the appropriate classes for your project is an experience-based effort. In other words, the authors made many mistakes in the beginning. To help you in designing classes, we have collected experience from our previous efforts.

So do what the authors did when they learned C++: find examples, copy, and paste!

This chapter introduces the thought process for creating classes, to answer questions such as these:

- How do I determine what is a class, and what is a method?
- How should I handle global functionality?
- What can inheritance do for me?
- What can operator overloading do for me?
- What does the C++ compiler create automatically?

Overview

Classes are fundamental to writing in an object-oriented language. But how do we decide what is a class? We have talked about thinking in terms of layers of abstractions. We have talked about roles and responsibilities. The next thing is to start to name the classes and their responsibilities. This is not as hard as it sounds. For one thing, you make classes as you feel they should be, and there is no right or wrong way. Let each class do what feels right to you. There will, of course, be some spirited team discussions. This is the topic of the first section of this chapter.

Once you decide on some classes, you can "wire up" instances of classes pretty much like you create and "wire up" modules in hardware. Unlike hardware, however, classes can have more "electricity." When designing hardware, you are restricted to connecting blocks through wires or signals, but with classes in C++, you have the ability, among other things, to have pointers to other class instances or call virtual methods. This is the topic of several sections.

This additional freedom is where the electricity comes in. This is good, because it helps you solve complicated verification problems. As with any technique, the challenge is to use the appropriate amount of electricity.

Not that everything has to be a class. C++ supports good old global scope functions, and for many situations, functions are appropriate. The section in this chapter on Global Services talks about various ways to package global functions in C++.

We end the chapter with a quick tour of some advanced class features of C++. We don't do this to make you concerned; rather, the purpose is to let you know that the language can solve pretty much any problem you will encounter.