thing amabob = {"wodget", -23};            //okay
Stock hot = {"Sukie's Autos, Inc.", 200, 50.25};   // NO!

The reason you can't initialize a Stock object this way is because the data parts have private access status, which means a program cannot access the data members directly. As you've seen, the only way a program can access the data members is through a member function. Therefore, you need to devise an appropriate member function if you're to succeed in initializing an object. (You could initialize a class object as just shown if you made the data members public instead of private, but making the data public goes against one of the main justifications for classes; that is, data hiding.)

In general, it's best that all objects be initialized when they are created. For example, consider the following code:

Stock gift;
gift.buy(10, 24.75);

With the current implementation of the Stock class, the gift object has no value for the company member. The class design assumes that the user calls acquire() before calling any other member functions, but there is no way to enforce that assumption. One way around this difficulty is to have objects initialized automatically when they are created. To accomplish this, C++ provides for special member functions, called class constructors, especially for constructing new objects and assigning values to their data members. More precisely, C++ provides a name for these member functions and a syntax for using them, and you provide the method definition. The name is the same as the class name. For example, a possible constructor for the Stock class is a member function called Stock(). The constructor prototype and heading have an interesting property—although the constructor has no return value, it's not declared type void. In fact, a constructor has no declared type.

**Declaring and Defining Constructors**

Let's build a Stock constructor. Because a Stock object has three values to be provided from the outside world, you should give the constructor three arguments. (The fourth value, the total_val member, is calculated from shares and share_val, so you don't have to provide it to the constructor.) Possibly, you just want to set the company member and set the other values to zero; this can be done with default arguments (see Chapter 8,