members. Thus, they must have distinct names; otherwise you end up confusing code like this:

shares = shares;

One common coding practice to help avoid such confusion is to use an m_ prefix to identify data member names:

class Stock
{
private:
    char m_company[30];
    int m_shares;
    ....

Using a Constructor

C++ provides two ways to initialize an object by using the constructor. The first is to call the constructor explicitly:

Stock food = Stock("World Cabbage", 250, 1.25);

This sets the company member of the food object to the string "World Cabbage", the shares member to 250, and so on.

The second way is to call the constructor implicitly:

Stock garment("Furry Mason", 50, 2.5);

This more compact form is equivalent to the following explicit call:

Stock garment = Stock("Furry Mason", 50, 2.5));

C++ uses a class constructor whenever you create an object of that class, even when you use new for dynamic memory allocation. Here's how to use the constructor with new:

Stock *pstock = new Stock("Electroshock Games", 18, 19.0);