• The class method definitions, which describe how certain class member functions are implemented

Roughly speaking, the class declaration provides a class overview, whereas the method definitions supply the details.

Listing 10.1 presents a tentative class declaration for a Stock class. (To help identify classes, we follow a common, but not universal, convention of capitalizing class names.) You’ll notice it looks like a structure declaration with a few additional wrinkles, such as member functions and public and private sections. We'll improve on this declaration shortly (so don't use it as a model), but first let's see how this definition works.

Listing 10.1 The First Part of stocks.cpp

```cpp
// beginning of stocks.cpp file
class Stock  // class declaration
{
  private:
    char company[30];
    int shares;
    double share_val;
    double total_val;
    void set_tot() { total_val = shares * share_val; }
  public:
    void acquire(const char * co, int n, double pr);
    void buy(int num, double price);
    void sell(int num, double price);
    void update(double price);
    void show();
};  // note semicolon at the end
```

You'll get a closer look at the class details later, but first let's examine the more general features. To begin, the C++ keyword class identifies this code as defining the design of a class. The syntax identifies Stock as the type name for this new class. This declaration enables us to declare variables, called objects, or instances, of the Stock type. Each individual object represents a single holding. For example, the declarations