(In OOP talk, that is the object to which the `topval` message is sent.) The problem is, what do you call that object? If you make the call `stock1.topval(stock2)`, then `s` is a reference for `stock2` (that is, an alias for `stock2`), but there is no alias for `stock1`.

The C++ solution to this problem is a special pointer called `this`. The `this` pointer points to the object used to invoke a member function. (Basically, `this` is passed as a hidden argument to the method.) Thus, the function call `stock1.topval(stock2)` sets `this` to the address of the `stock1` object and makes that pointer available to the `topval()` method. Similarly, the function call `stock2.topval(stock1)` sets `this` to the address of the `stock2` object. In general, all class methods have a `this` pointer set to the address of the object invoking the method. Indeed, `total_val` in `topval()` is just shorthand notation for `this->total_val`. (Recall from Chapter 4, “Compound Types,” that you use the `->` operator to access structure members via a pointer. The same is true for class members.) (See Figure 10.4.)

**Figure 10.4.** `this` points to the invoking object.