Adventures in Functions.

Thus, the prototype would look like this:

```c
// constructor prototype with some default arguments
Stock(const char * co, int n = 0, double pr = 0.0);
```

The first argument is a pointer to the string that is used to initialize the company character array class members. The `n` and `pr` arguments provide values for the `shares` and `share_val` members. Note that there is no return type. The prototype goes in the public section of the class declaration.

Next, here's one possible definition for the constructor:

```c
// constructor definition
Stock::Stock(const char * co, int n, double pr)
{
    strncpy(company, co, 29);
    company[29] = '\0';
    shares = n;
    share_val = pr;
    set_tot();
}
```

This is the same code that we used for the `acquire()` function. The difference is that a program automatically invokes the constructor when it declares an object.

**Caution**

Often those new to constructors try to use the class member names as arguments to the constructor:

```c
// NO!
Stock::Stock(const char * company, int shares, double share_val)
{
    ...
}
```

This is wrong. The constructor arguments don't represent the class members, they represent values that are assigned to the class