

THE USE OF "WRAPPERS" TO PREVENT MULTIPLE DECLARATIONS OF C++ OBJECTS

THE PROBLEM

When you start compiling object-oriented program, you can get things in header files multiply defined. This is because you include the header files in the class implementation and also in the driver. As the number of objects and the interaction between them increases, this problem can also increase.

THE SOLUTION

The way to prevent this is to put a protective "wrapper" around the header file that only includes it if it hasn't been included already.

The header file should look like this:

```
~~~~~  
#ifndef Name_H  
#define Name_H  
  
//  
// The header file declarations go here  
//  
  
#endif
```

~~~~~  
Name\_H is the name of the header file with a \_H appended to it. This name MUST be unique for this header file.

It works pretty much like it looks. If Name\_H isn't defined, the compiler defines it and includes the header. The next time the compiler runs into the header, Name\_H will already be defined and it will be ignored.