



CS 211 *Chapter 11*

Separate Compilation and Namespaces

OOP:

- One of the main tenets of Object Oriented Programming is Encapsulation – Information Hiding.
- The implementation should be separated from the interface (class definition and function/operator declaration) into different files – Header Files, and Implementation Files.



Header File:

The interface file is always a header file which ends with the suffix .h

It includes class definition and all function declaration.

Although all private member variables and functions are part of implementation, since the class definition can only happen once, thus, these member variables and function declaration also included in the header file.



Implementation File:

Any program files, as well implementation files, which uses the class must include the header file.

If there is no implementation file, any program that uses the class which only has the header file will compile, but won't run the program.

The implementation file has the same name as the header file, but has the suffix .cpp

It contains the definitions of the member functions and overloaded operators.

Compile this implementation file every time after changes.

Driver File/Application File:

The program file which uses the class. It also must include the .h header file

To run the program, you must first compiling the application file and the implementation file.



C Preprocessor:

- **Compiler's preprocessor puts the macro identifier on a list, and modifies a source code file where the identifier to replacement before handing it over to the compiler.**
- **User it to check whether it has happened before (include files directly into other files)**
- **define constants or inline code.**



Example:

```
#ifndef MEASURE_H
#define MEASURE_H
    #define Default_Value 0
    #include <iostream>
    using namespace std;
    class Measure {
    ...
    Measure():foot(Default_Value), inch(Default_Value){
};

#endif
```

