**一，开始**

**1，声明和定义**

"声明"只是声明这个符号的存在，"定义"则是把一个符号完完整整地描述出来。

一个符号在整个程序中可以被声明多次，但却要且仅要被定义一次。

**2，头文件的作用**

头文件不用被编译，把所有的函数声明全部放进一个头文件中，当某一个.cpp 源文件需要它们时，可以通过一个宏命令 "#include" 包含进这个.cpp文件中，从而把它们的内容合并到.cpp文件中去。

**3，#include**

#include是一个来自C语言的宏命令，在预编译的时候就会起作用。它的作用是把它后面所写的那个文件的内容，完完整整地，一字不改地包含到当前的文件中来。

**4，头文件内容**

头文件的内容在多个 .cpp 文件中得到了编译。通过"定义只能有一次"的规则，我们很容易可以得出，头文件中应该只放变量和函数的声明，而不能放它们的定义。

如果写上 int a或者 void f() {}这样的句子，那么一旦这个头文件被两个或两个以上的 .cpp 文件包含的话，编译器会报错。

**5，三个例外**

1，文件中可以写const对象的定义。

2，头文件中可以写内联函数的定义

3，头文件中可以写类（class）的定义。

**6，综述**

.h文件中能包含：

类：

成员数据的声明，但不能赋值

静态数据成员的声明和定义，但不建议，只是个声明就好。

成员函数的声明

内联函数的定义

非类：

非类成员函数的声明

常数的定义：如：constint a=5;

静态函数的定义

**7，头文件中的保护措施**

#ifndef SALES\_DATA\_H

#define SALES\_DATA\_H

XXXXXX

#endif

使用#pragma once可以获得相同的效果。但是，这种方式不支持跨平台！