

一、 全面了解 C++ 头文件中可以包含的内容。列举出相应的代码。

头文件由三部分内容组成：

1. 头文件开头处的版权和版本声明
2. 预处理块
3. 函数、类和结构声明等。

头文件应该只用于**声明**对象、函数声明、类定义、类模板定义、typedef 和宏，而**不应该**包含或生成占据存储空间的对象或函数的**定义**。

举例：

```
1. extern void Foo1();    //function declaration
2. extern int a1;         //object declaration
3. class A;               // forward declaration
4. class B
5. {
6.     private:    A* pa;
7. };
```

```
1. #include "header.h"
2. int a1 = 0;            //defination 正确，在头文件中声明了
3. static int32_t a3 = 0; //正确，是静态的
4. static void Foo2()
5. { //...}               //正确，是静态的
6. void Foo1()            //正确，在头文件中声明了
7. { //...}
```

二、 上网搜索，研究 C++ 头文件的包含顺序。

1) 列出《Google C++ 编程风格指南》中的观点，主要解决什么问题？

《Google C++ 编程风格指南》中指出 C++ 头文件应使用下面的顺序：

1. C 标准库
2. C++ 标准库
3. 其它库的头文件
4. 你自己工程的头文件。

这样做主要解决了 C++ 头文件的可读性和避免隐含依赖性等问题。

2) 列出《C++编程思想》中的观点，主要解决什么问题？

头文件被包含的顺序是从“最特殊到最一般”：

1. 在本地目录的任何头文件
2. 我们自己的所有“工具”头文件
3. 第三方库头文件
4. 标准 C++ 库头文件和 C 库头文件。

主要解决了如下问题：保证.h 文件的组成部分不被它自身解析,这可以避免潜在的使用错误。因为被自身解析缺乏明确提供的声明或定义。在.c 文件的第一行包含.h 文件能确保所有对于构件的物理界面重要的内部信息块都在.h 中（如果的确是缺少了某些信息块，一旦编译这个.c 文件时就可以发现这个问题）。