1. 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？ 2)列出《C++编程思想》中的观点，主要解决什么问题？ 3)什么是头文件包含依赖，举例说明。 二、上网搜索，全面了解C++头文件中可以包含的内容。列举出相应的代码 完成时间：2周
2. 在Google式的编程风格中，include采取的顺序是：（例如本文件名为“fun.cpp”）
3. 本文件对应的文件头：“fun.hpp”
4. 使用的C语言系统文件头：“stdio.h”
5. 使用的C++语言系统文件头：“iostream”
6. 其他相关库的.h文件头：“myfun.h”
7. 本项目的其他.h文件头：“fund\_2.h”

这一推荐文件头顺序的意义在于减少重复的互相依赖，保障每个文件头都能独立地通过编译；其效果是保障了每一个文件头都是在对应.c或.cpp文件中第一个通过编译的。这样也可以显著减少重复的文件头出现。

在同一类型级别的文件头，建议按照字母顺序排序。

1. 在《Thinking in C++》中，include采取的顺序是从特殊到一般，亦即以上顺序中的2-5倒过来。
2. 本地目录的任何头文件：“fun.h”；
3. 自用的所有工具头文件：“fund\_3.h”；
4. 第三方库头文件：“myfun.h”；
5. 使用的标准C++文件头：“iostream”
6. 使用的标准C文件头：“stdio.h”

其目的在于，保证每个文件的组成部分不被它自身解析，可以避免潜在的使用错误，而被自身解析缺乏明确的声明或定义。如果某一个头文件不被自身解析，那么我们可以马上找到这一错误。

1. 头文件包含依赖，一般指头文件中包含其他头文件，导致如果对这个头文件编译的话，一定要同时编译其包含的头文件。由于在代码界面层次上这种行为比较隐蔽，但是如果头文件出现不包含或未定义的情况，又会导致不可预料的问题，故而造成漏洞和危险。

比如有三个头文件a.h b.h c.h，a.h里面有#include "b.h",b.h里面有#include "c.h", c.h里面有#include "a.h"，那就会造成文件的循环依赖，后果是什么呢？

比如有个文件a.c，上面有#include "a.h"，那在a.c文件编译之前，预处理程序就会不断的把这三个头文件的内容复制过来，超过了一定的数量，就会导致“头文件数太多”的编译错误。

解决方法呢，当然就是常见的#ifndef...#define...#endif组合了。不过要把前两个写在头文件的开头（一定是开头），最后一个写在最末尾。

这样的话，第一次展开a.h b.h c.h的时候就已经定义了宏，到了c.h中的#include "a.h"时候，遇到了#ifndef，由于这个宏在上一次展开时已经定义了，所以这部分就跳过去了。也就是每个头文件最多只在每个源文件里面包含一次。

但是即使编译链接没有问题，循环依赖也会降低开发效率，为什么？因为文件都在依赖，比如某一天，要改变a.h的一部分内容，然后所有依赖于a.h b.h c.h的文件都得重新编译...链接；所以现在的C++有“前向声明”的技巧可以缓解这个问题。（缓解并不是解决。）而JAVA运用的import机制就很好的解决了这个问题，真正实现了“实现与声明相分离”这个目标。

当然，#pragma once可以起到同样的效果。

1. 头文件可以包含的内容

C++中头文件应该包含内容如下：

（1）类的定义、extern变量、函数的声明；

（2）在编译时已知的const对象的定义（用常量表达式初始化），因为当const变量用常量表达式初始化时，可以保证所有变量具有相同的值；

（3）Inline函数，内联函数可以直接当文本展开，不需要调用；

（4）在类的定义中一般可以使用引用或者指针可以声明的数据成员，就不要用对象完成， 因为如果是指针或者引用，只需要一个类型声明；但如果是对象就需要该类型的定义。 这个时候要为声明式和定义式提供不同的头文件。

（5）头文件中必须使用完全限定的标准库名字，不能使用using声明或者using指示，以 防重复包含。也就是只能用std::cin>>a,而不能写成using std::cin；