C++标准程序库中的所有标识符都被定义于一个名为std的namespace中。 由于namespace的概念，使用C++标准程序库的任何标识符时，可以有三种选择：  
1、直接指定标识符。  
　　例如std::ostream而不是ostream。完整语句如下： std::cout << std::hex << 3.4 << std::endl;  
2、使用using关键字。  
　　using std::cout; using std::endl; using std::cin; 以上程序可以写成 cout << std::hex << 3.4 << endl;  
3、最方便的就是使用using namespace std  
　　例如： 　　#include<iostream> 　　#include<sstream> 　　#include<string> 　　using namespace std; 　　这样命名空间std内定义的所有标识符都有效（曝光）。就好像它们被声明为全局变量一样。那么以上语句可以如下写: cout << hex << 3.4 << endl; 因为标准库非常的庞大，所以程序员在选择的类的名称或函数名时就很有可能和标准库中的某个名字相同。所以为了避免这种情况所造成的名字冲突，就把标准库中的一切都放在名字空间std中。但这又会带来了一个新问题。无数原有的C++代码都依赖于使用了多年的伪标准库中的功能，他们都是在全局空间下的。 所以就有了<iostream>和<iostream.h>等等这样的头文件，一个是为了兼容以前的C++代码，一个是为了支持新的标准。 命名空间std封装的是标准程序库的名称，标准程序库为了和以前的头文件区别，一般不加".h"