**c++简介**

1.面向对象；

2.检查严格；

3.操作符重载；除了基本数据类型，自己定义的类型也可以进行+、-、\*、/、运算。

4.支持异常处理机制；

5.支持泛型编程；泛型指的是，模糊的类型。如对int型数据进行求最大值的函数，并不适用于float型的数据，需要另外再写一个函数。如果支持泛型编程，则可以直接使用一个函数，实现所有类型的操作。

C语言将变量看成状态，而函数经常看成操作/过程（从标准库函数可以看出，如printf()）。

而C++则把它们看成是对象，如屏幕，键盘。cout << endl;

#include <iostream> //没有.h后缀，有时候提供.h版本头文件。一般没有。

std::cout << "hello,c++!" << std::endl它可以自动的区分出变量的类型。

比printf()函数，更加聪明一些。

编译器：g++

也可以使用gcc，但是要加上-lstdc++。默认使用的是c标准库。

扩展名：.cpp/.cc/.C/.cxx windows后缀不区分大小写。有的编辑器如SI不支持.cc。

也可以使用.c，但是要求加上-x c++。 -x是知道语言的参数。

兼容c:可以使用printf()函数，但是包含的头文件可以是用c++的<cstdio>

很多想stdio.h的文件，C++都另外起了一个名字。c。。。。

**名字空间**

什么是名字空间----WHAT？

-名字空间定义

-名字空间合并

-声明定义分开

怎样用名字空间----HOW？

-作用于限定符

-名字空间指令

-名字空间声明

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

命名空间::,作用域。

namespace abc

{

正常的函数的声明。void bar();

也可以把定义写到外面，但是需要使用void abc::bar()

也可以定义函数。

}

void abc::bar()

using namespace adc

之后abc空间内的所有声明均可以使用。

部分使用命名空间的内容：

using abc::foo；

编译器一行一行的进行检查。看到namespace后知道声明过。

-------------------------------------------------

无名命名空间。

想使用命名空间外的全局变量，如：

float g\_balance =0;

namespace abc

{

float g\_balance =111;

void show(void)

{

return g\_balance;

}

void withdraw()

{

g\_balance+=10;

::g\_balance+=11; //使用的是命名空间abc的外面的g\_balance.

}

}

-----------------------------------------------

命名空间可以嵌套。

namespace ns1

{

namespace ns2

{

namespace ns3

{

void foo(void){};

}

}

}

使用，ns1::ns2::ns3::foo();

那么可以使用别名，来简化书写。

namespace n3=ns1::ns2::ns3;

n3::foo();

--------------------------------------------

两个命名空间有相同的函数，且在文件头已经

using namespace了呢？？？