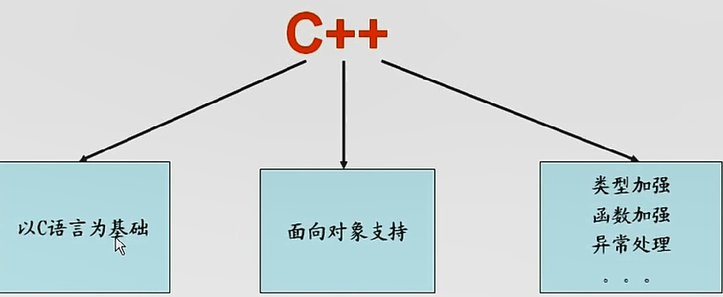
### 一.C++与C的关系

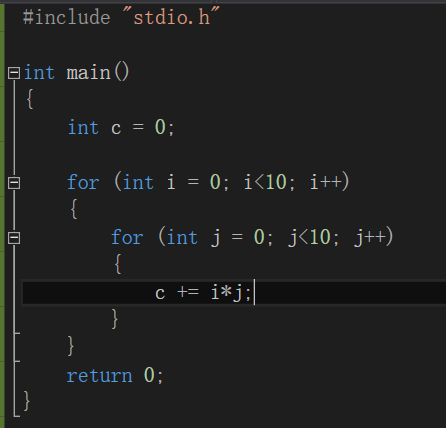
C++继承了所有C的特性

C++是对C的加强

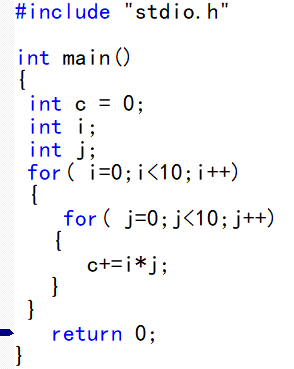


二.C++对C的加强

1.C++更强调语言的使用性，所有变量都可以在需要的时候再定义



如上面的程序在C程序中不能编译通过，C++中可以

C程序必须这样

才能编译通过

2.register关键字

C语言中无法获取register变量地址

C++中可以获取，register对变量的声明变得无效

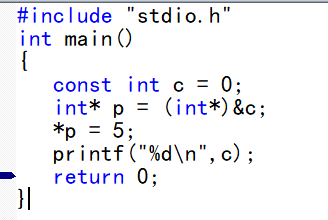
3.C语言中，重复定义多个同名的全局变量合法

C++不允许。

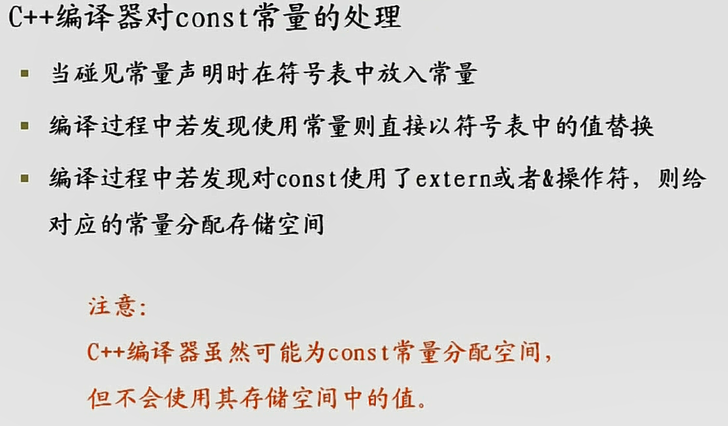
C语言中多个同名的全局变量最终会被链接到全局数据区的

同一个地址空间上

4.const



在C语言中会输出5，C++会输出0

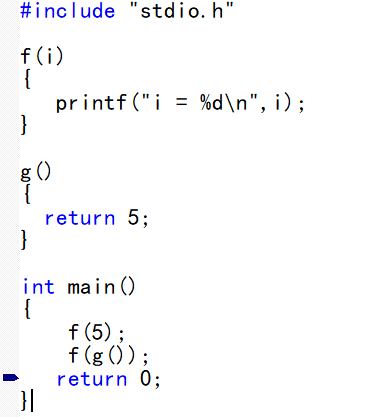


C语言中的const常量是只读变量，有自己的存储空间

4.struct

C语言的struct定义了一组变量的集合，C编译器不认为这是一种新类型

C++中的struct是一个新类型的定义声明



在C语言中能编译通过，输出i=5

i=5

它默认缺省的是为int类型

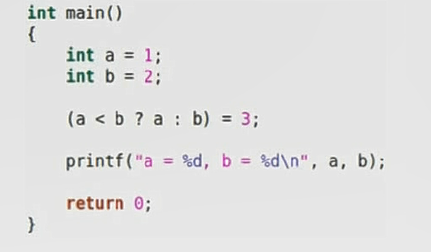
5.bool类型

C++增加了bool类型，bool可取的值只有true和false

理论上bool只占用一个字节

编译器会将非0值转换为true，0值转换为false

6.?:运算符

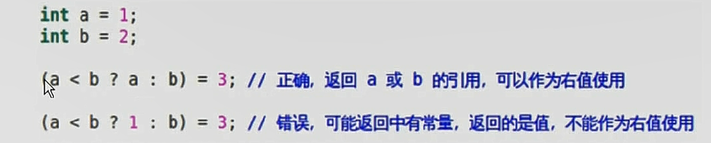


c++中会通过编译并输出a=3,b=2

因为返回的值是a，所以将a赋值为3.

当三目运算符的可能返回都是变量时，返回变量引用

当三目运算符的可能返回中有常量时，返回的是值



7.引用

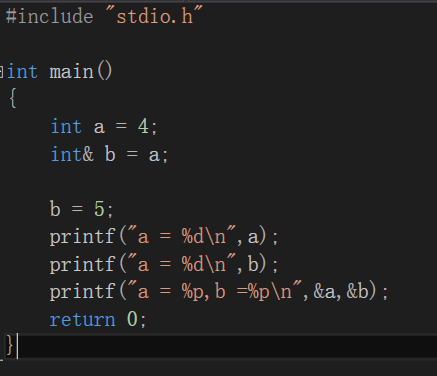
变量名：即一段实际连续存储空间的别名

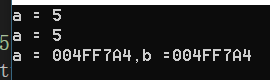
程序中通过变量来申请并命名存储空间

通过变量的名字可以使用存储空间

引用可看做一个已经定义变量的别名

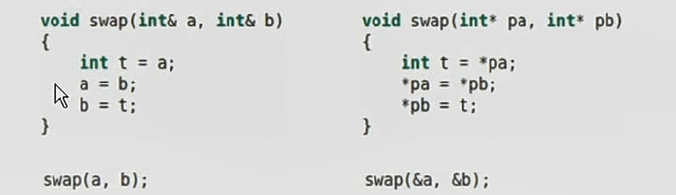
语法：Type &name = var;



输出：

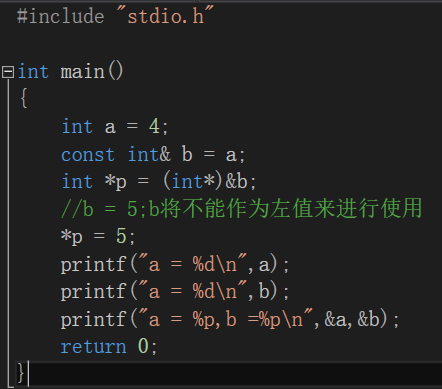
普通引用在声明时必须使用其他的变量进行初始化

引用作为其他变量的别名，因此可以在一些场合可以代替指针



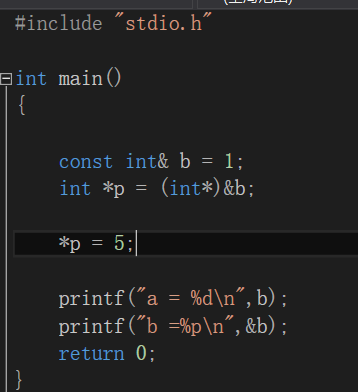
引用作为函数参数声明时，不进行初始化

const引用



const引用让变量拥有只读属性

当使用常量对const引用进行初始化时，C++编译器会为常量分配空间，并将引用名作为空间的别名



引用在C++中的内部实现是一个常指针，Type& name 等价于 Type\* const name;

C++编译器在编译过程中使用常指针作为引用的内部实现，因此引用所占的空间大小与指针相同

当函数返回值作为引用时：

若返回栈变量（局部变量）

不能成为其它引用的初始值

不能作为左值使用

若返回静态变量或全局变量

可以成为其它引用的初始值

即可成为右值使用，也可作为左值使用

