**1 hello world**

#include<iostream>

#include<cmath>

using namespace std;

int main()

{

cout << "hello world" << endl;

system("pause");//诸塞功能

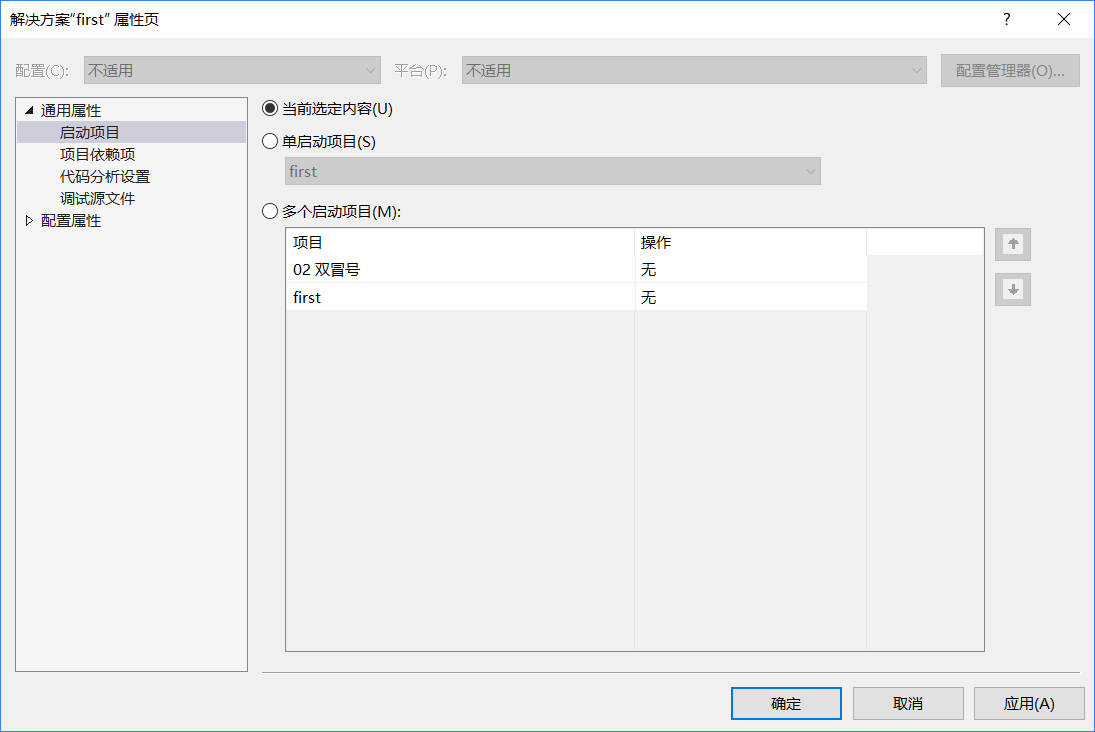
return 0;

}

**2 作用域**

出现c4996错误，需要用到宏

#define\_CRT\_SECURE\_NO\_WARINGS



就近原则输出100

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

using namespace std;

int atk = 200;

void test01()

{

int atk = 100;//局部变量

cout << "攻击力："<<atk<<endl;

cout << "攻击力：" << ::atk << endl;//输出全局

双冒号前什么也没写就是全局作用域

}

int main() {

test01();调用

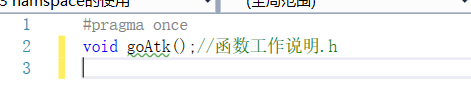
system("pause");

return 0;

}

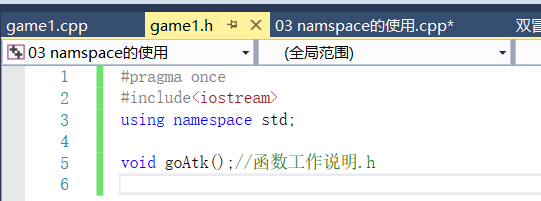
3. namespace使用

头文件用于函数工作说明

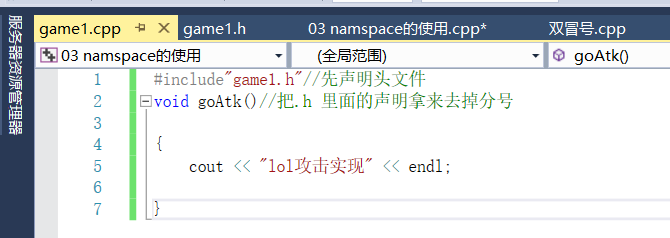


头文件使用

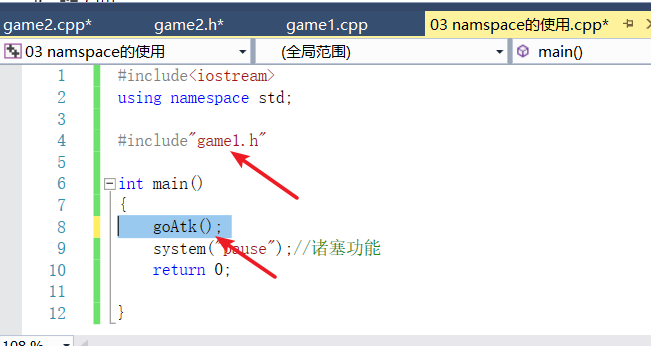
.h文件用于说明函数

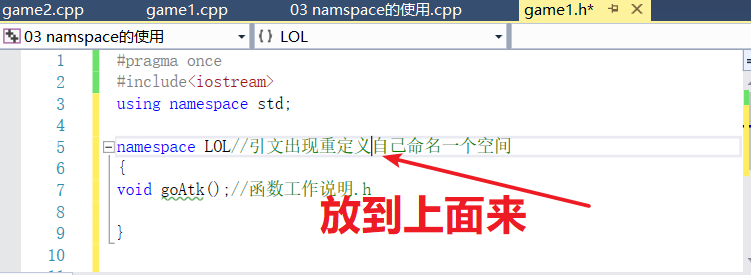


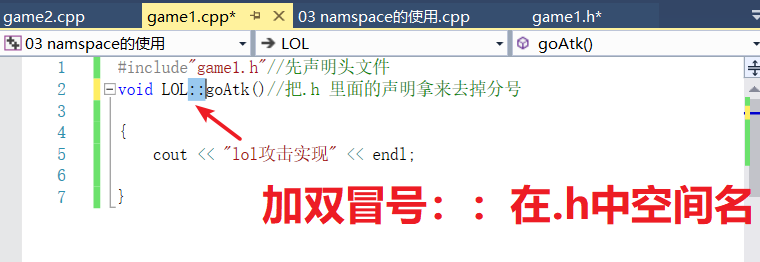
然后在.cpp可以作为函数的实现



在其他cpp文件中包含头文件和函数名称即可调用







Namespace 用来解决名字冲突的问题

//1.命名空间下可以放函数，类，变量，结构体，类

//2. 命名空间必须定义在全局下

//3. 命名空间可以嵌套

//4.命名空间开放可以随时往原先的命名空间添加内容

//5. 无名命名空间，相当于写了静态变量，只能在当前文件夹中使用

//6. 命名空间可以去别名

//using 声明避免二义性

//写了uisng 声明后 下面这行代码说明以后看到sunwukongID使用kingglory下的

//但是又有就近原则出现二义性

//using KingGlory::sunwukongID;

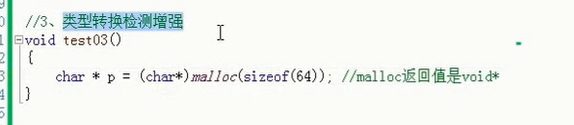
07:c++对C的加强

1. 全局变量的检测增强，C++中会发现重定义
2. 函数检测增强，参数类型增强，函数调用参数增强

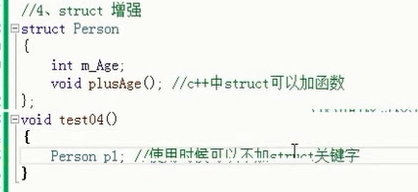
在C++中定义了函数，还要返回值

1. 类型检测增强

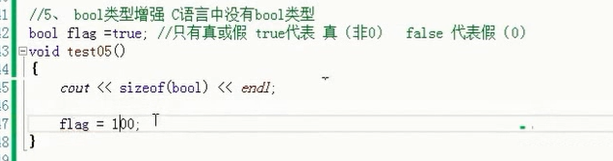
Malloc返回值是viod\*



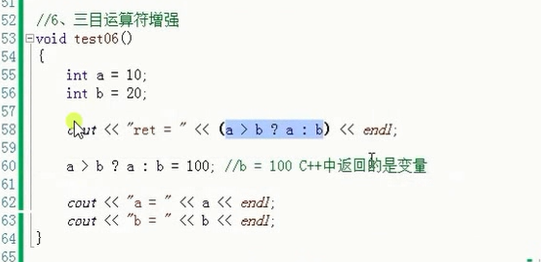
1. Struct 增强，C++可以加函数



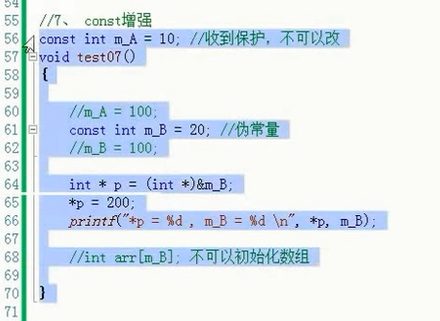
1. Bool 类型增强 C语言没有bool



1. 三目运算符增强

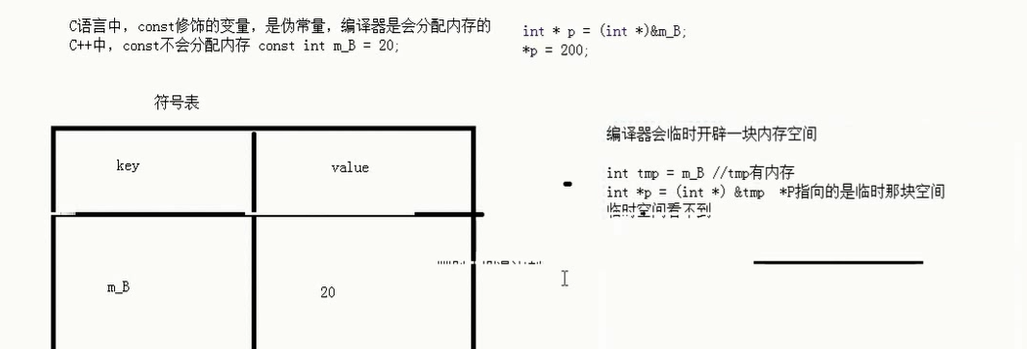


7.

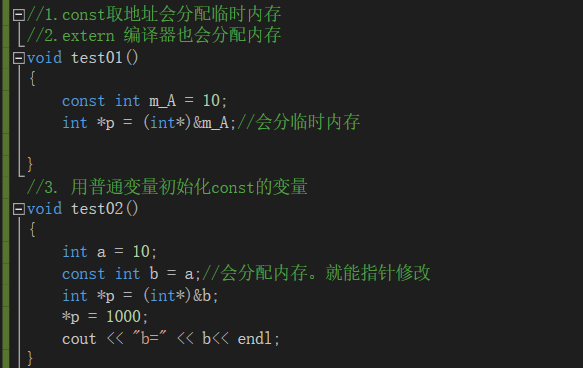


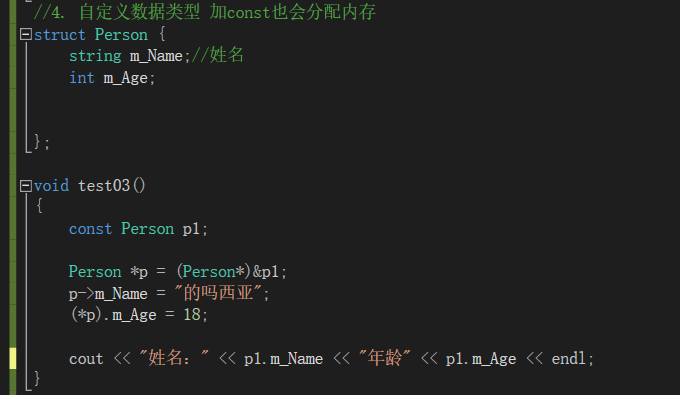
8.

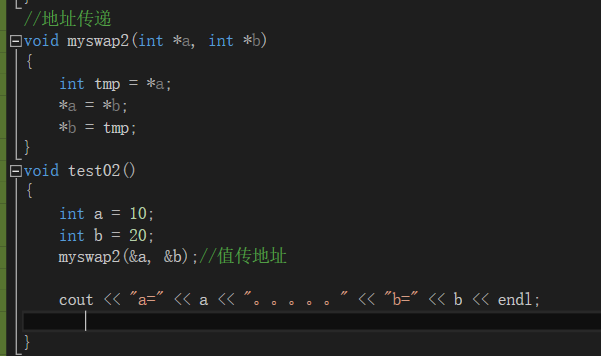
取地址的时候编译器会临时开辟一块内存空间

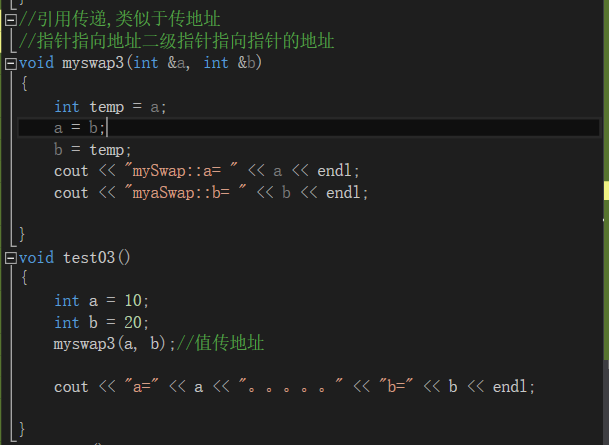


9. C++访问const的时候要加一个extern



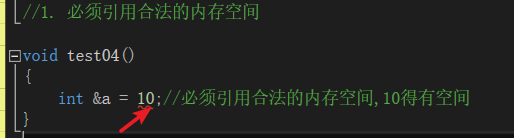




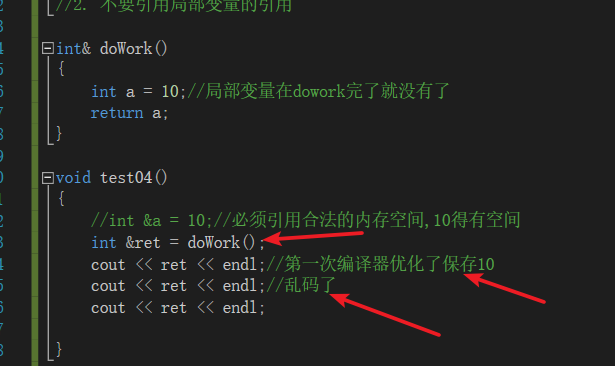


//引用注意事项

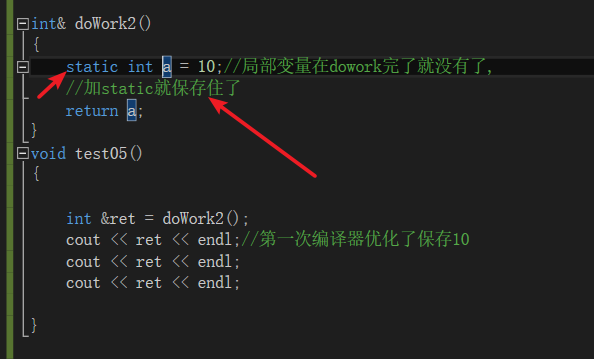
//1. 必须引用合法的内存空间



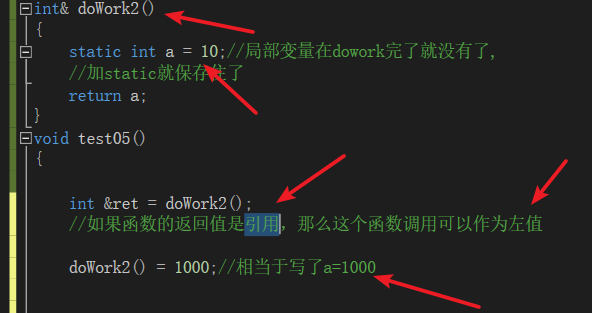
//2. 不要引用局部变量的引用

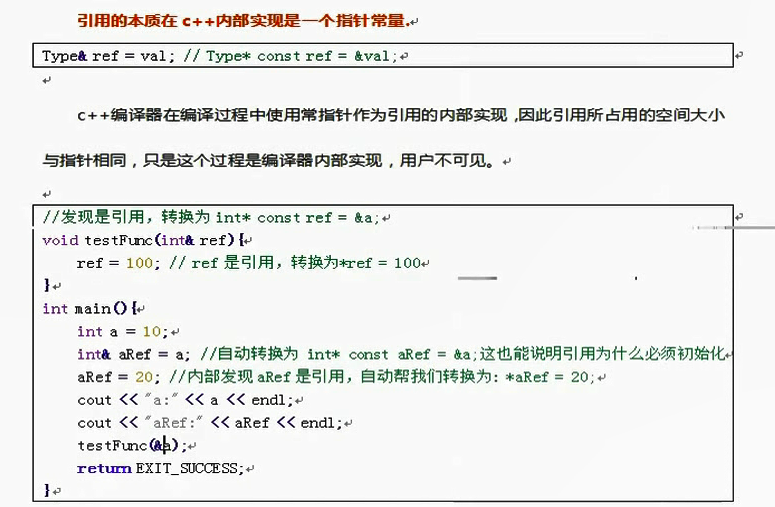


解决办法

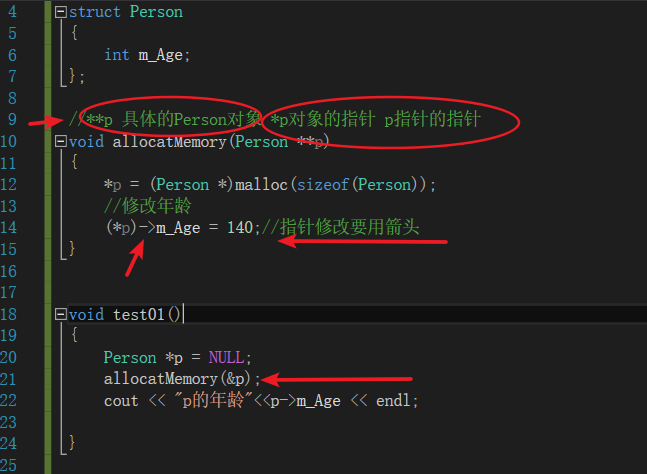


函数可以作为左值

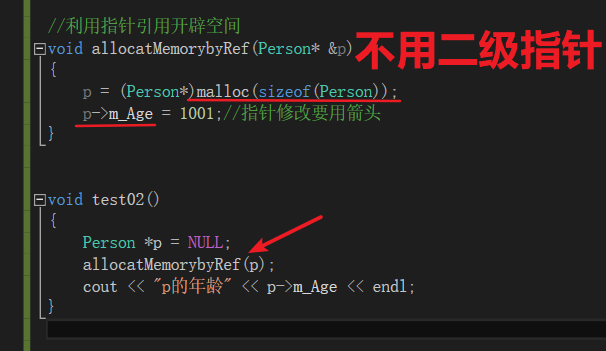




//\*\*p 具体的Person对象 \*p对象的指针 p指针的指针



//利用指针引用开辟空间



常量引用

