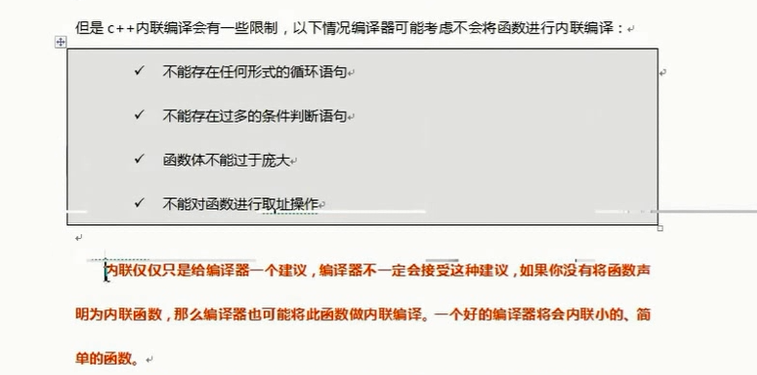
封装，继承，多态

类

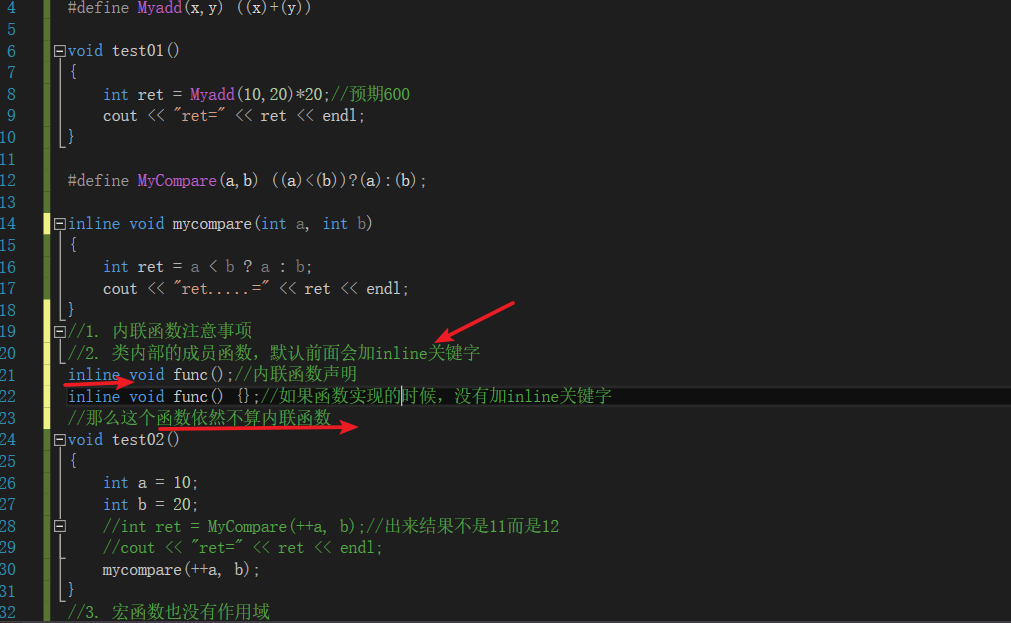


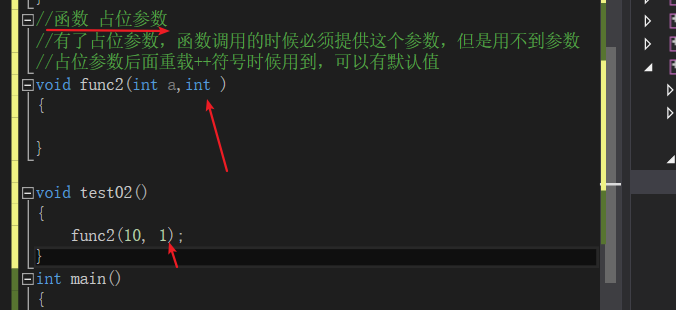


03 内联函数宏函数的缺陷

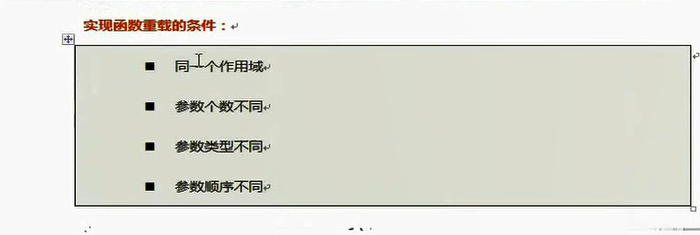


内联函数解决了红缺陷的问题

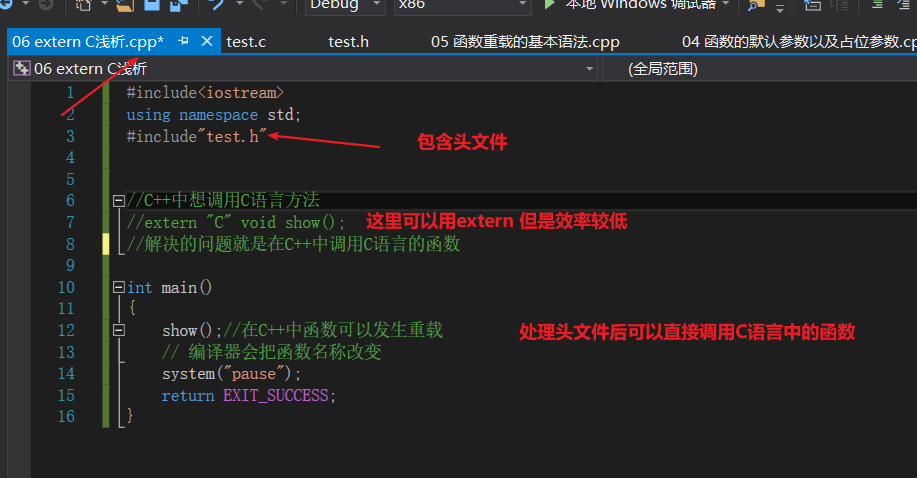


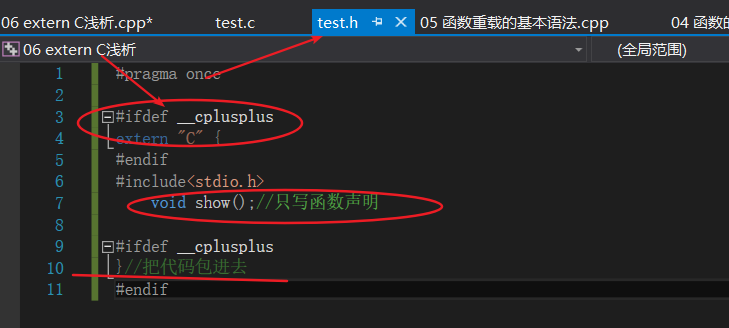


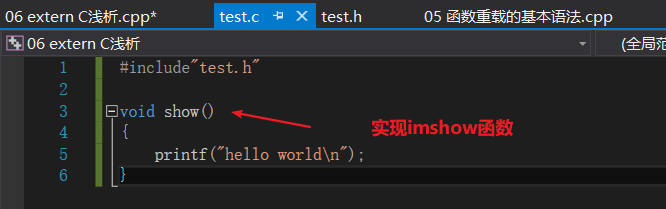
函数重载

引用的时候const 可以作重载的条件

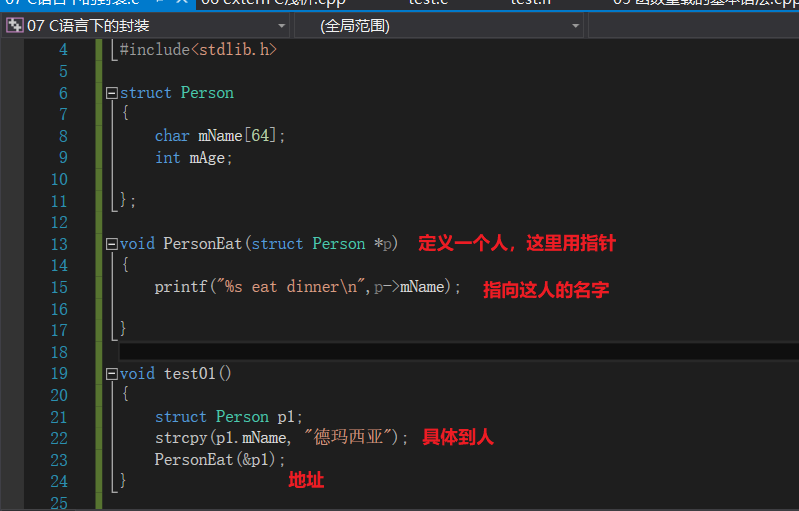
Extern C 浅析







封装



protected://保护权限，类内部可以访问（当前类的子类可以访问）类外不可以访问

private: ://保护权限，类内部可以访问 类外不可以访问(子类不可以访问)

建议就是把变量设置为private型

class Person

{

public:

void setAge(int age)

{

if (age < 0 || age>100) {

cout << "impossbile" << endl;

return;//return 后就不能赋值了

}

m\_Age = age;

}

//年龄只能读

int getAge()

{

return m\_Age;

}

//读姓名

string getName()

{

return m\_Name;

}

//写姓名

void setName(string name)

{

m\_Name = name;

}

//只写的情人

void setLover(string lover)

{

m\_lover = lover;

}

private:

int m\_Age=0;//年龄只读

string m\_Name;//共有读写

string m\_lover;//情人只写

};

void test01()

{

Person p1;

p1.setName("老王");

cout <<"p1's name=" << p1.getName() << endl;

cout << "p1's age" << p1.getAge() << endl;

p1.setAge(10);

p1.setLover("常老师");

}