//类中被称为接口的是public的类成员

//友元函数通过访问类对象来访问所有成员

//也就是说非成员函数想要访问private,用友元函数+对象.成员即可

用 class 定义一个类时，数据成员和成员函数的默认访问权限是private

!!!!!!!!!!重载运算符复习一下几元运算符那些东西！

几个量参与运算，就是几元，成员重载仅1参数

// 注意： . .\* :: ?: sizeof 五大不能重载

// 2：

// 重载运算符只能用成员重载的有= () [] ->,其他都用友元就行

1．设有类A的对象Aobject，若用成员函数重载前置自增表达式，那么++Aobject被编译器解释为（ A ）。

（A）Aobject.operator++() （B）operator++(Aobject)

（C）++(Aobject) （D）Aobject :: operator++()

4．设有类A的对象Aobject，若用友元函数重载后置自减表达式，那么Aobject--被编译器解释为（ B ）。

（A）Aobject.operator-- () （B）operator-- (Aobject,0)

（C）-- (Aobject,0) （D）-- (Aobject,0)

友元放在括号里，成员.

// 构造函数不可以是虚函数，但析构可以；

. 虚析构函数的作用是（ C ）。

（A）虚基类必须定义虚析构函数 （B）类对象作用域结束时释放资源

（C）delete动态对象时释放资源 （D）无意义

构造函数可以重载，但析构不行;构造函数可以设置默认参数，但析构不行

复制构造函数在对象初始化，函数有类参或者返回类时调用

构造函数在创建类的对象时候调用！

析构函数虚函数可以用在有派生类的类中

3．在创建派生类对象时，构造函数的执行顺序是（ ）。

（D）基类构造函数—对象成员构造函数—派生类本身的构造函数

成员齐全，才能启航。

例如：

#include<iostream>

using namespace std;

class Base1

{ public :

Base1( int i )

{ cout << "调用基类Base1的构造函数:" << i << endl; }

};

class Base2

{ public:

Base2( int j )

{ cout << "调用基类Base2的构造函数:" << j << endl; }

};

class A : public Base1, public Base2

{ public :

A( int a,int b,int c,int d ):Base2(b),Base1(c),b2(a),b1(d)

{ cout << "调用派生类A的构造函数:" << a+b+c+d << endl; }

private :

Base1 b1;

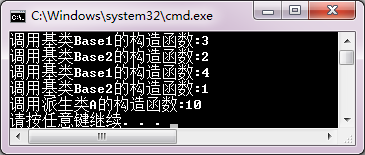
Base2 b2;

};

int main()

{ A obj( 1, 2, 3, 4 );

}



// 父类指针可以指向子类对象

// 调用重载函数不属于运行时的多态，属于编译多态，模板函数也是编译多态

虚函数是运行时多态，需要对象指针调用

// 一个类模板定义了静态数据成员，则( A )。

//A．每一个实例化的模板类都有一个自己的静态数据成员。

//B．每一个实例化的对象都有一个自己的静态数据成员。

//静态数据成员是属于类的，同类所有对象相同

// 但也说明不同模板类静态对象不共享！！

静态数据成员：

当处于private:不可以被类的对象或者类名调用，只可以用static函数调用

当public都可以

静态成员由于全类共享，所以收到所有对象的操作影响，要结合全局去看结果

必须在类外初始化，受private控制符作用

注意：友元函数、静态函数没有this指针

// 1. 右所示类格:class X<-class Y<-class Z ;类X中有数据成员int a。

//根据以下函数注释的编译信息,分析int X::a的访问特性,

// class Y对class X和class Z对class Y的继承性质。

//void Y : : funY() { cout << a << endl; }//正确

//void Z : : funX() { cout << a << endl; }//错误

//void main()

//{

// Xx; Yy; Z z;

// cout << x.a << endl;//正确

//

// cout << y.a << endl;//错误

//

// cout << z.a << endl;//错误

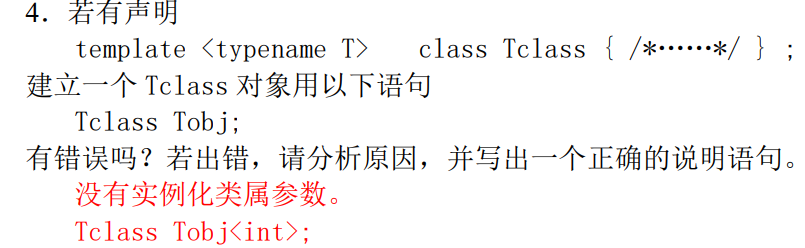
//}

// int X::a是class X的public数据成员, class Y为protected继承class

// X, class Z为private继承class Y。

//分析：protected继承不可通过对象名访问，但可以函数访问

注意，当出现构造函数和赋值函数一样时，定义两个的原因是，一个初始化，一个改值。



对于类模板，实例化T后（模板类）再调用

函数模板的实例化由编译器实现

一个类定义中，只要有一个函数模板，这个类就是类模板

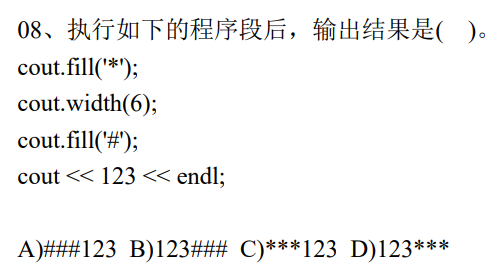
类模板的成员函数都是函数模板，类模板实例化后，成员函数也随之实例化关于类模板与友元叙述错误的是（ A ）。

（A）一般函数可以声明为类模板的友元

（B）函数模板可以声明为类模板的友元

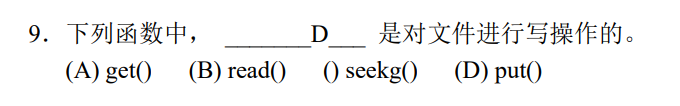
（C）一个模板类可以声明为另一个类模板的友元

（D）一个模板类的友元只能是模板



选A，为什么123在右边？？？？？？？？？？？？？？？？？？？？

默认输出从右



Istream库全是读操作，输入流

read():读（指定长度）

get():读（指定分隔符）

Getline()

Cin>>:读，提取运算符

//ios::in以输入方式打开与ifstream连用;

Ostream库全是写操作，输出流

put():写一个

write():写一行，（指定长度）

Cout<<:写，插入运算符

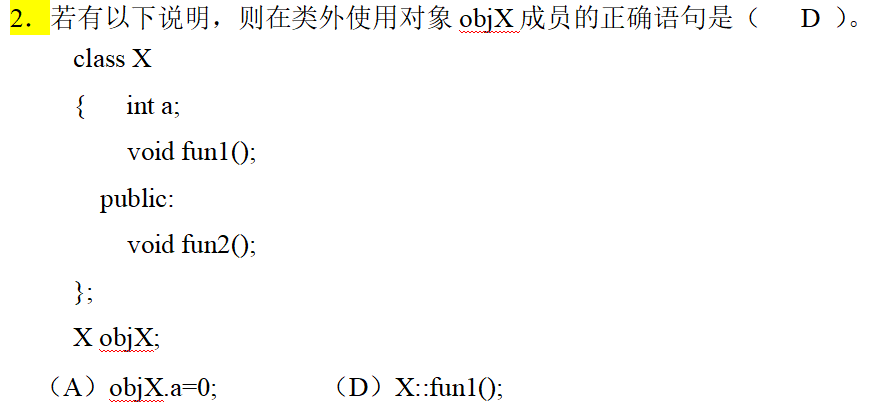
// ios::out以输出方式打开与ofstream连用

// （指的是以CPP视角看数据，流出Output就是写文件里面，流入Input就是导入CPP）

Ios::app追加打开文件

// 抽象类引用

// 首先，如果想要调用抽象类函数，必须引用，因为抽象类不能创建对象



原因：默认为private，private只能通过成员函数调（当然friend也可以)

抽象类

题看到P65了