第二次选择题

4月3日8:17 4月11日 20:19

1.B重载运算符是对用户自定义类定义的，

D双目运算符，在成员函数里的运算符重载，只需要提供一个参数；如果运算符重载是全局函数，需要提供两个参数

双目运算符包括：

+ (加)，-(减)，\*(乘)，/(除)，% (取模)

1. 除了流运算符以外，移动构造函数也可以避免由于对象复制而造成的额外开销

B不必要是const Test&,const可以不加

C ++作为后缀的时候，需要一个int类型作为哑元参数，如果没有哑元，就是前缀运算符

D 运算重载一般有两种方式:【1】全局函数运算符重载；【2】成员函数运算符重载（两者的参数不同）！

3.A若用户已经定义了其他构造函数，编译器将不会隐式合成默认构造函数

B全局对象在main调用前被构造，但的确在main结束前不会被析构

C引用不能更改指向对象，也不能指向空

D清除对象占用的资源是无条件的，不需要任何选项。因此，析构函数没有参数，且只有一个（即清除方式唯一）。//但是在{}里加入一些额外的操作还是可以的

4.

A main函数构造类A的对象应该是 A a;或者A a=A();A a();是声明了一个没有参数的函数

B 编译器会隐式地合成了一个B的默认构造函数，并在B b处调用

C 如果一个class里面有其他class类型作为其成员变量，那么调用成员变量的析构函数的原则：先执行自己的析构函数，再调用成员变量的析构

D 此处’c’先被转换成int型值，然后调用构造函数A(int i)

默认转换的方法：先把c转化成ascii码，再转化成int。这样的非法调用在编译的时候会显示warning，但不会error

如果重载了以char为参数的构造函数，就会调用这个构造函数了

5.A 友元函数可以是全局函数

B 可以声明别的类的成员函数，为当前类的友元

C 析构函数和构造函数都可以被声明为别的类的友元函数

D B的成员函数可以访问A的成员

6.A常量对象不能调用非常量成员函数

B常量成员函数永远都不能修改对象里的成员变量，所以不能调用非常量成员函数（实际上编译器不能辨认非常量成员函数是否进行了修改成员变量的操作

C 静态变量在不同的文件中不共享。include相当于把static int v复制到了三个cpp文件中，在每个文件里都代表不一样的东西

D 大多数成员的确如此，但是int是例外；int是可以进行类内本地初始化的

1. B错：静态数据成员在类实例化对象前已分配内存空间

//注：全局变量的初始化早于静态成员变量

8.A member1 是const类型，只能在构造函数初始化列表里初始化；而member2在不同的Test对象中都一样，可以在Test实例化前初始化

C常量成员函数可以访问和修改静态数据成员！

D常量对象可以调用常量成员函数和静态成员函数。（静态成员函数只能修改static成员）

第二次作业第二题 by zhaochen20

4月2日13:47；4月2日19:21；

这个题的要求，除了（1）这些地方，别的都不可以添加代码

Example\* e[10];//这里是创立了十个example类型的指针，e本身是example\*\*，指针数组的表头

e是指针数组，因而调用e[i]对象的元素时需要使用e[i]->getData();



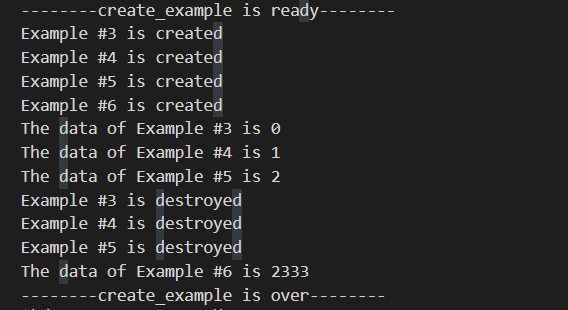
注意到1号对象在main之前就创建了，因此实际上是在（7）这个位置创建的。

如果在main之前就构造了对象，会先调用构造函数；如果你把对象建构在main的第三行，则会先输出再构造。但是先声明（这是先声明还是先定义，我去查查，待会更新）在main以外，则会先调用构造函数；

getData();//注意getData的奇怪的大小写

cout << "--------main\_function is over---------\n";//"\n"才是换行，不要反了，我错了好几遍呜呜

//查一下如何类似列表的for循环，待会更新



注意到这个地方，传进去了n=3，实际上构造了四个，所以还需要手动再在相应的位置构造一个对象。而且，这个对象在函数体结束了没有被析构，故而这个对象是个static。

注意到析构的顺序，正常情况下是先创建的后析构（我去查查，不太确定），但是这里既然反向析构了，那就是一个for循环哇。

实际上关于这个题的析构顺序：最后是按照7261的顺序析构的：6是在主函数调用的函数里函数里创的static，所以在主函数全部结束了才析构；7最后创建且就是一个主函数的对象，先析构，接下来析构主函数创建的对象2，之后析构6，最后析构全局变量。

注意到全局实际上需要一个变量来记录创建的对象的个数：故而需要一个static int在private里面。然后这个static int在.h里申明，在.cpp里赋值。

静态的成员变量不属于对象，而是属于类，就是说这个变量对类的所有成员共享

赋值的方式是：

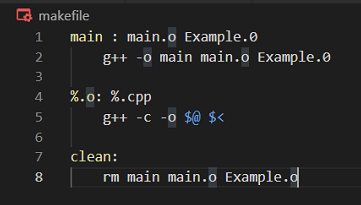
int Example::index=0;//Example.index和Example::index区别？我待会查查，貌似是对某个具体对象应该用a.data这种，但是对于一个类的函数，比如构造函数，应该用Example::Data才能调用private。

这个类需要有三个成员数据：一个static int记录全局，一个data，一个whatever，记录自己的编号。

void create\_example(int n)

注意到create是个类外函数，不能访问private，但是他又输出了private数据，那么可以想见getData这个public成员函数的作用，让create访问getdata，getdata访问data。这就是OOP的精髓.jpg

Ps.这次我make file都能有花样错误，猜猜这玩意儿咋错的，呜呜呜



第二次作业第三题

4月3日19:22；4月4日8:53 4月4日20:181

1. 要做的事情包括：



1. map的题目：是常量不能重载——中括号要重载两次，const函数是不能改变参数的，只有const成员能够调用const函数。定义了一种类型，const对象调用函数会自动调用const函数。
2. []的重载问题

【1】重载位置：==,[],(),->只能通过成员函数来重载！不能用全局函数重载。

【2】重载返回类型：

如果返回类型是引用，则数组运算符调用可以出现在等号左边，接受赋值，即

Obj[index] = value;

如果返回类型不是引用，则只能出现在等号右边

Var = Obj[index];

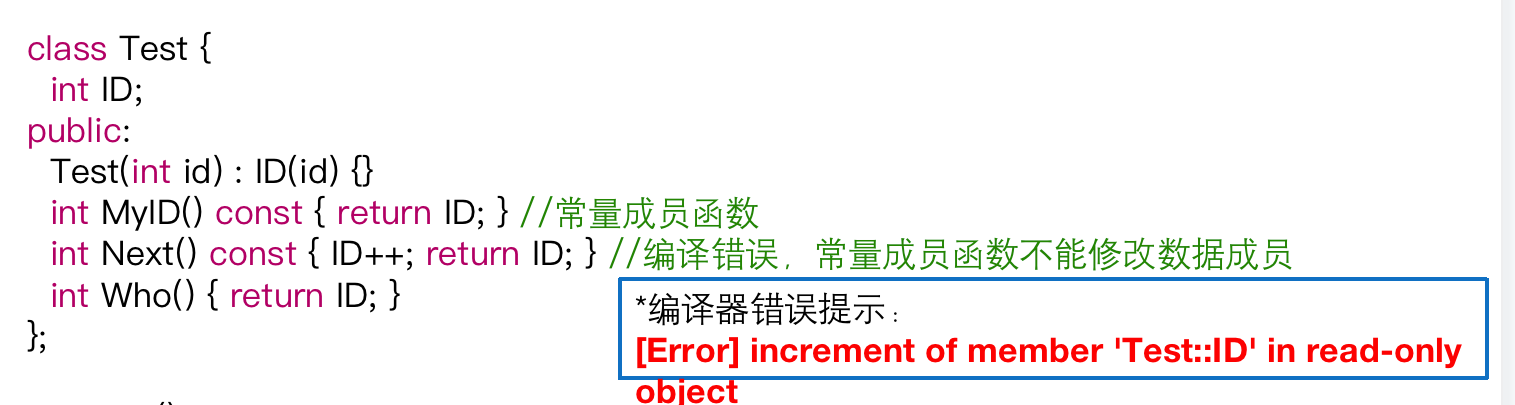
这道题中出现了对数组运算符调用赋值的操作，所以那些返回类型一定是【引用】！

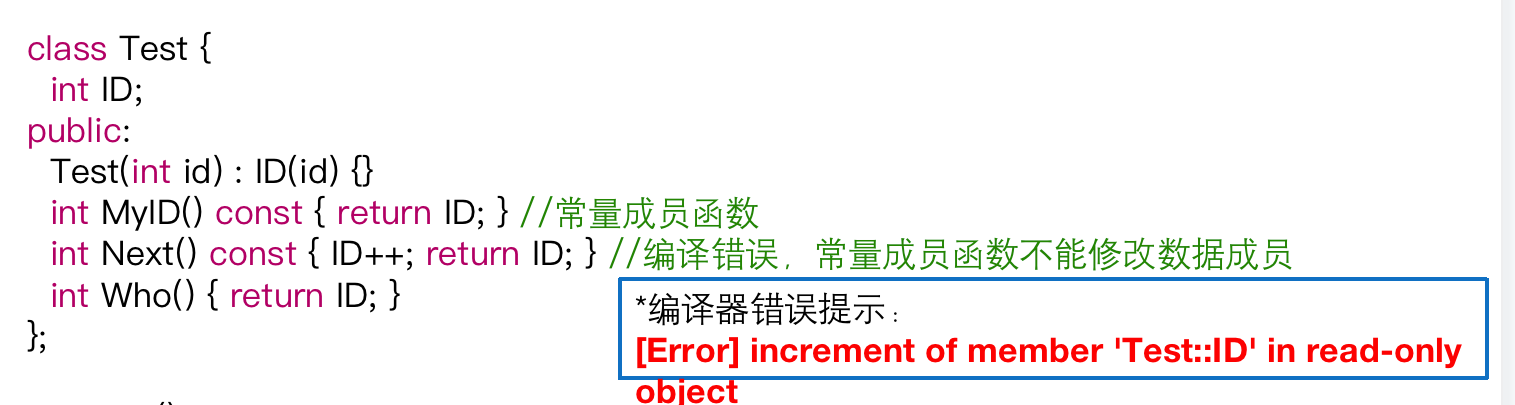
注意区分【常量成员函数】和【函数返回值是常量】：

常量成员函数：Typename Func() const {return type;}//注意这个const写在()后面，{}前面！

常量成员函数不能修改它所在对象里的其他成员。

在这道题里，cmap是常量对象，它只能调用【常量成员函数】（Typename Func() const {return type;}）。因而cmap[key]对应的重载函数一定是const类型





返回值是常量：const Typename\*Func(){};

要求函数返回值只能读不能写的时候，可以用const修饰函数返回值（如const int\* func() {…}），此时函数返回值的内容（或其指向的内容）不能被修改

lzy的题其实有很多奇怪的细节，比如大小写。有的函数会在中间的某个字母突然大写，以及，所有的cpp和h的开头都是大写。

没有参数的函数，需要加小括号。

 const Map & cmap = map;

cmap是map的引用（别名），但是cmap是const，所以cmap不能被修改。也就是，修改map时cmap会同步被修改；但是cmap本身是const类型，不能被修改。

cout << map[key] << endl;//map[key]返回的是引用，这样你修改map[key]的时候，map里面的对象就被修改了，这就是‘能读能写’的含义（而不只是能读）

实际上，即使[]的重载函数不是常量成员函数，得到的值虽然可以修改，但是如果不加引用，仍然无法修改pair数组里的原值。但是，不加引用非常量成员函数的返回值本身可以修改，而常量成员函数连本身都不能被修改，如此一来这句就不能实现了

            cout << cmap[key] << endl;//重载cmap的[]和map的[]；实际上因为cmap是const对象，只能调用const成员函数，所以你写两种重载函数，一个给const用的const重载，一个给一般成员用的一般重载

    data[sz-1].reset(x,0);

oop的精髓：不能直接调用private，先调用public函数，再用public函数调用private

注意这道题要写【析构函数】！

原因：隐式定义的析构函数不会delete【指针】成员（在这里是data），可能造成内存泄露

但是注意:由于我们不知道n，不能用一个循环把所有pair析构掉

可以直接用delete []data，这样就直接把data指向的整个数组都删掉了。编译器自动调用pair的析构函数

new xxx[n]就要对应用delete [] xxx，不会出错

【Remark：】

int Pair::n\_create = 0;

int Pair::n\_free = 0;

这两个变量可以不管它！

还有几个弱智的错误点，555

【1】在.h文件里【声明】函数不能用{}，一旦有了{}就是定义了，表示这个函数什么都不干

【2】在调用无参函数的时候不要忘记()。比如是gerVal()而不是getVal

第二次作业第四题

4月5日上午6:36

本次作业最复杂一题：大概的任务：在缓冲区读取进入n个人的信息，并把这些信息处理为五种数据成员。之后按照数据成员进行相应的检索。题干本身并不复杂，但是由于技术操作比较繁琐，所以需要保持头脑清醒。

注意到你main.cpp里可以看出，School对象里面包含有n个Student对象，且你是把student一一add member进去的，那么school（注意区分大小写，本题中school与student都该是大写，我这里小写了图个方便）里面至少包含三个数据成员：

一个int存储学生的数目，一个student型的指针用于动态分配student（记得要析构函数把这里delete掉，并且考虑到查找不到时我们要输出“Not Found”，所以我在这里分配了一个空学生，找不到就返回这个空学生，如果是空学生就输出找不到），以及一个用来给add member计数的num；

现在开始设计Student需要的数据成员。

我的设计如下：

    string name;

    string home;

    string id;

    int age;

    string date;

//这里用string是出于：如果你用int的话，用atoi（注意是atoi函数，用来把string转化为int，不是auto类型界定）会把“0031”转化为31，这样在输出的时候需要补充0

    //但其实补充0并不麻烦，要不乖乖写的时候用int叭，这样避免咱俩查重；（好奇怪，dbq）

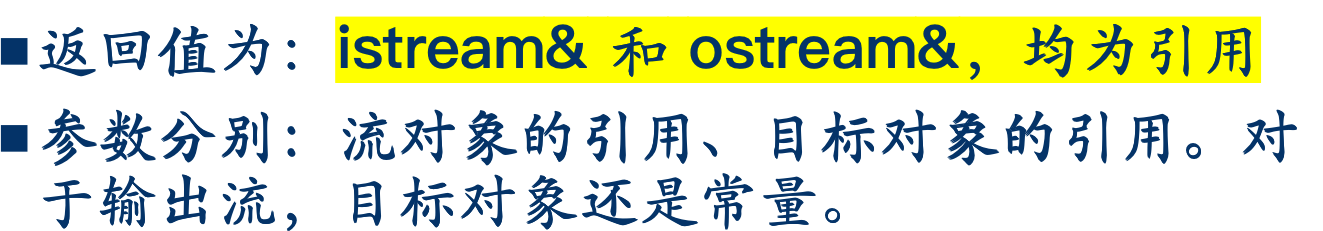
    //另外，就算在string里面 321也是大于123的，所以可以用string直接比大小

接下来的任务是重载对于School类型的>>（注意到我们重载实际上是对于某个类型重载，比如我们没法直接输入一个School类的对象，但是可以直接输入一个int类的对象，因此你重载的仅仅只是School的>>。并且，我们重载的是in，不是cin，这个待会说明。）

（>>和<<的重载不能在类内实现。它们是全局函数）

（想要用>>和<<输入哪个类型的对象，就把重载函数设为谁的友元（friend，重载函数声明写在这个类型的public里头，前面加上friend。函数的定义则写在这个类型的cpp文件里。但是注意，这时在cpp文件里不需要加Classname::。）





这个重载就要完成对于输入区的切割。

（注意，这里是不能用scanf函数的。原因是：scanf是从标准输入流读取数据，但是getline是从你给定的流里面读取，来自助教）——呃其实我还是没懂……

istream& operator>> (istream& in, Student& dst)//in可以理解为你输入的内容，这个内容是istream类型

这个是申明格式，当然这是在Stuent.cpp里写的；那么怎么分割字符串呢，这里用了一个getline函数，他是一个istream类型的成员函数，

char a[100];

in.getline(a, 99, '-');

表示a这个字符数组从输入缓冲区（in的内容）的起点开始读取并且储存，最多读99个字符，读到’-’就把’-’去掉，然后停止。

//这里其实涉及到了输入的原理：c++里的输入是有一片缓冲区，然后你把一堆乌七八糟的玩意儿输到这个缓冲区里面

    //然后比如 string s; cin>>s;就是把这个缓冲区的所有东西，一个一个按照顺序读入到s里面；

    //而in.getline()这个函数部分，注意到我的写法是先有一个a[100]，然后开始一个一个地读入，读入的最大长度为99，读入的终点是"-"符号

    //注意到，getline函数是不含尾部的，你的'-'保留在缓冲区，且'-'打头的那些乌七八糟的玩意儿还可以再读入；那我们接着读入；

//实际上还有别的读取方式，比如遇见一个'-'停留一次的那种，但是比较麻烦；

    //另外，由于只有名字的字符长度是不确定的，读完名字后，可以按照后面固定的长度剪字符串即可。

注意到一个坑的地方。（啊啊啊这个太有用了。

OJ给的输入实际上是"12\nasdsad-asdasd\nsfafa"，因此你第一个人读入了\n;最后你把这个\n也当成了名字，但为啥第二个人没有发生这件事？因为你的getline会把终止符去掉

平时不会发生这件事是因为我们默认都是以\n为终止符的，这里用’-’做终止符就会出错。所以处理的方法可以有：先开一个char a;in.getline(a,1)（dbq，我不知道\n是一个字符还是俩，貌似是一个，贝贝待会去问问？）把终止符读走。或者我的弱智办法是特判了一下起点是不是\n。

char a[100];

    in.getline(a, 99, '-');//OJ给的输入实际上是"12\nasdsad-asdasd\nsfafa"，因此你第一个人读入了\n;最后你把这个\n也当成了名字，但为啥第二个人没有发生这件事？因为你的getline会把终止符去掉

    string nam(a);

    string x;

     if(a[0]=='\n')

     {

         for(int i=1;i<nam.length();i++)

            {x += nam[i];}

             dst.name=x;

     }

     else

     {

         dst.name=nam;

     }

建议不要用特判，防止查重！！！

接下来同样操作按照数据类型，读取缓冲区再不断把char数组转化为你想要的东西就可以了。

值得注意的是关于char数组转化为int

 dst.age=atoi(e);//我一开始用age=atoi(e);会报错，为啥？因为你的age是对象的；也就是非静态成员引用必须与特定对象相对；

重载了>>之后记得要返回 in；重载了<<也要返回out；

重载完了建议加入一个输出，看看读取的对不对

 //cout <<"\n\n" << dst.name << "-" << dst.home << "-" << dst.age << "-" << dst.date <<"-" << dst.id <<"\n\n";

return in;//虽然这里只有一个输入，但是为了使他满足>>a>>b这种套娃输入，就return一个in

全程用的都是in和out，而不是cin和cout，因为你在重载out，你希望你的out对于所有的输出流都适用，那么不能写cout。因为还有fout这些

//如果写了cout，没有问题，但是没法给fout这些用，因此写out习惯更好

//这里参数的意义：cout << asdfsad << sadf; out就是你的cout，src就是你的asdfsad，然后调用了函数，返回了一个新的ostream&，从此开始套娃

写到这里其实很晕的，回到主线，我们现在处理add member，这个就是把输入的Studnet s加到你动态分配生成的那个student数组里。

接下来是文本高潮：

            int sid;

            cin >> sid;

            cout << tsinghua[sid]

这几行，实际上需要完成三件事：重载School的[]、重载Student的<<、然后重载Student的逻辑比较符号;

[]需要重载三次（对三种不同的类型）；

Student& School::operator[](const int x)//这里采用记录下标的方式，避免了多重循环

//注意到id的查询和name查询都是惟一的，所以不用重载>符号

以int为例，这里的格式：返回值类型（Student&）+类的名称::+后面一堆，

实际上我在这里翻了好几个弱智操作，经常把类型打在最前面，实际上是把返回值类型写在最前面。

{   int j=n;

    string t = to\_string(x);//这里完成了int转为string

    for(int i=0;i<School::n;i++)

    {

        if (st[i].id==t&&j==n)

        {

            j=i;

            break;

        }

    }

    return st[j];

}

这里其实蛮简单的，就是把int转化为string，再比较记录，返回了st[j]注意到返回值为引用是不能返回临时变量的，j是临时变量。但是你的st[j]不是临时变量。

对于省份“一个大写的char”转为string，

    int j=n;

    string t;

    t += x;//这里完成了char转为strin

省份需要重载student的>符号。

注意logic问题，//最晚回校的学号最大，意思是先最晚，再学号最大；

完成了这些，最后把你返回的Student&的信息打印出来，如果是空学生就打印找求不到。

打印就是重载<<，格式PPT上有。记得return out就可以了。