**第三章课后习题感受和新得**

第三章课后习题给我的感受就是每个知识点都要完全掌握，每个题都会有一些自己生疏的知识点，比如说在类声明中，成员函数的原型的参数表中可以不说明参数的名字，而只说明参数类型；而隐性内联函数和显性内敛函数更是让我对内联函数有了更深刻的认识。再有就是构造函数的形式二，用new运算符动态建立对象的方式，让第一章的new和delete和构造函数联系起来，拓展了它的功能，也是我对他有了一个新的认识。

以下是归纳的知识点

1.比较特殊的成员类型：protected.

保护成员在本类中和private类型的成员作用一模一样。区别在于保护成员可以由本类的派生类的成员函数访问，但是私有成员在其派生类中无法访问。

1. 成员函数的定义

成员函数如果在类外定义，那么函数声明的参数表可以只写参数类型，但是在类外定义时必须给出参数名。

3.内联成员函数

（1）隐式定义内联成员函数，直接在类内定义的成员函数

（2）显式定义内联成员函数

用关键字inline声明（在声明和定义的语句句子开头加inline），此时可以在类内声明，类外定义。声明、定义都要在句首加inline.

1. 内联函数

内联函数的代码会在编译时插入到每一个调用它的地方。这种做法会提高运行效率。但是只有很简短的代码才实用。

2.对象（成员访问，初始化方式）

1.对象成员的访问方式：

例如对对象d中成员year的访问

d.year (\*p).year p->year

2.对象的初始化

（1）

class complex

{

public:

double real; //数据成员是public时才能使用这种方法

double imag;

};

complex c={1.1,2.2};

（2）用构造函数（作用：为对象分配空间，进行初始化）

例如：

...

class complex

{

public:

complex(int a,int b)

{real=a;imag=b;}

private:

double real;

double imag;

};

int main()

{

complex A(1,2);//或者complex \*p=new complex(1,2)也可以，不过不需要它时要用delete p;释放

//空间

}

1.构造函数也可以重载。但是调用无参的构造函数时应该 complex A，不能complex A()

2.定义构造函数后，系统将不再提供默认构造函数。所以对于带参数的构造函数，在定义对象时必须给构造函数的形参传值，否则构造函数将不会被执行。

3.but...构造函数可以设置默认参数解决这个问题，例：complex(int a=0,int b=0) 在定义对象时如果complex A;此时默认值和形参结合，不会出现 2 的情况。注意默认参数必须在构造函数声明时设定。一个程序只能设置一个默认构造函数。：complex(int a=0,int b=0)和complex()只能有一个，不然执行complex A;时会出错。一般不要同时使用函数的重载和有默认参数的构造函数:complex(int a=0,int b=0);complex(int a)；complex s(1)此时会出现二义性。

4.与构造函数对应的有析构函数。没有参数，不能被重载，且一个类只能有一个析构函数。

（3）成员初始化列表

这种方法不在函数体内用赋值语句进行初始化，而是在函数首部实现的

接着上面的例子：..

...

public:

complex(int x,int y)：real(x),imag(y)

{ }

...

//或者构造函数在类体外定义时候成员初始化列表也可以

complex::complex(int x,int y)：real(x),imag(y)

{ ...}

注意：数据成员是按照在类中声明的顺序初始化的，和在成员列表中的排列顺序无关。

const修饰的和引用类型的数据成员都不能用构造函数直接初始化，只能用成员初始化列表。因为const修饰的数据只能初始化，不能赋值，构造函数中赋值的过程；

优点：主要是性能问题，对于内置类型，如int, float等，使用初始化类表和在构造函数体内初始化差别不是很大，但是对于类类型来说，最好使用初始化列表，为什么呢？使用初始化列表少了一次调用默认构造函数的过程，这对于数据密集型的类来说，是非常高效的。

3.对象数组和对象指针

1.对象数组：complex A[3]={undefined

complex(1,1),complex(2,1),complex(2,3)};

2.对象指针：指向对象的指针也可以指向对象数组。

3.this指针

每当创建一个对象的时候，系统就把this指针初始化为指向该对象，即this指针的值是当前调用成员函数的对象的首地址。

例如：执行A.show()就相当于this->show()

4.string类

1.string在C++标注库中已经声明，可以直接定义string类的对象。使用string类对象是程序开头必须加上#include<string>

2.初始化 string str1("china")或者string str1="china"都可以。

3.使用string类可以直接进行基本的运算。