通过对第四章的学习，我了解了派生类的基本概念，派生类的构造函数与析构函数，调整基类成员在派生类中的访问属性的一些其他方法以及学习了多重继承。学会了派生类对象中包含基类对象，因此派生类对象在创建时，除了要调用自身的构造函数进行初始化外，还要调用基类的构造函数初始化其包含的基类对象。因此，程序中任何能够生成派生类对象的语句，都要说明其包含的基类对象是如何初始化的。如果对此不做说明，则编译器认为基类对象要用无参构造函数初始化——如果基类没有无参构造函数，则会导致编译错误。在执行一个派生类的构造函数之前，总是先执行基类的构造函数。和封闭类说明成员对象如何初始化类似，派生类说明基类对象如何初始化，也需要在构造函数后面添加初始化列表。在初始化列表中，要指明调用基类构造函数的形式。派生类对象消亡时，先执行派生类的析构函数，再执行基类的析构函数。在 C++ 中，派生可以是多层次的。例如学生类派生出中学生类，中学生类又派生出初中生类和高中生类。总之，类 A 派生类 B，类 B 可再派生类 C，类 C 又能派生类 D，以此类推。这种情况下，称类 A 是类 B 的直接基类，类 B 是类 C 的直接基类，类 A 是类 C 的间接基类。当然，类 A 也是类 D 的间接基类。在定义派生类时，只写直接基类，不写间接基类。派生类沿着类的层次自动向上继承它所有的间接基类。派生类的成员包括派生类自己定义的成员、直接基类中定义的成员，以及所有间接基类的全部成员。当派生类的对象生成时，会从最顶层的基类开始逐层往下执行所有基类的构造函数，最后再执行自身的构造函数；当派生类对象消亡时，会先执行自身的析构函数，然后从底向上依次执行各个基类的析构函数。继承是Ｃ＋＋语言的一种重要机制，它允许在已定义的类的基础上产生新类。

我们称从已定义类产生新类的过程称为派生。已存在的用来派生新类的类为基类，又称父类。由已存在的类派生出的新类称为派生类，又称为子类。如上例，从哺乳动物派生出狗，哺乳动物是父类，狗是子类。

在Ｃ＋＋语言中， 一个派生类可以从一个基类派生，也可以从多个基类派生。

从一个基类派生的继承称为单继承；从多个基类派生的继承称为多继承，由两个基类一个风扇类和一个台灯类派生出一个风扇台灯类。继承的处理从定义基类开始，基类定义了所有对于派生类来说普通的属性和方法，派生类从基类继承所需的属性和方法，且在派生类中增加新的属性和方法。继承重用父类的代码，专注于为子类编写代码。

一.派生类的定义格式

1． 单继承的定义格式:

class <派生类名>:<继承方式><基类名>

{

<派生类新定义成员>};

解释:

(1) <基类名>是已定义类的名称。 <派生类名>是新定义的一个类的名字，它是从<基类名>中派生的。

(2) 派生类按指定的<继承方式>从基类中派生的,<继承方式>有常用的如下三种:

public 表示公有继承

private 表示私有继承

protected 表示保护继承

2 多继承的定义格式

class<派生类名>:<继承方式1><基类名1>,<继承方式2><基类名2>,......

{

<派生类新定义的成员>

}

多继承与单继承的主要区别从定义格式上看，主要是多继承的基类多于一个。