在C++中，继承就是在一个已存在的类的基础上建立一个新的类。

一个新类从已有的类那里获得其已有特性，这种现象称为继承。

从已有的类产生一个新的子类，称为类的派生。

一个派生类只从一个基类派生称为单继承；

一个派生类有两个或多个基类称为多继承。

而声明派生类的方式为：

class 派生类名：[继承方式] 基类名

其中的继承方式有三类：公有、私有、保护三类继承方式

而其访问规则为：

若继承方式是public，基类成员在派生类中的访问权限保持不变，也就是说，基类中的成员访问权限，在派生类中仍然保持原来的访问权限；

若继承方式是private，基类所有成员在派生类中的访问权限都会变为私有(private)权限；

若继承方式是protected，基类的共有成员和保护成员在派生类中的访问权限都会变为保护(protected)权限，私有成员在派生类中的访问权限仍然是私有(private)权限。

基类成员在派生类中的访问属性



当一个派生类具有多个基类时,这种派生方法称为多继承。

多重继承的构造函数的执行顺序与单继承构造函数的执行顺序相同,也是遵循先执行基类的构造函数,再执行对象成员的构造函数,最后执行派生类构造函数体的原则。处于同一层次的各个基类构造函数的执行顺序,取决于声明派生类时所指定的各个基类的顺序,与派生类构造函数中所定义的成员初始化列表的各项顺序没有关系。析构丽数的执行顺序则刚好与构造函数的执行顺序相反。

当引用派生类的成员时,首先在派生类自身的作用域中寻找这个成员,如果没有找到,则到它的基类中寻找。如果一个派生类是从多个基类派生出来的,而这些基类又有一个共同的基类,则在这个派生类中访问这个共同的基类中的成员时,可能会产生二义性。为  了解决这种二义性,C++引人了虚基类的概念。

虚基类的初始化与一般的多继承的初始化在语法上是一样的,但构造函数的调用顺序不同。在使用虚基类机制时应该注意以下几点:

(1)如果在虚基类中定义有带形参的构造函数,并且没有定义默认形式的构造函数,则  整个继承结构中,所有直接或间接的派生类都必须在构造函数的成员初始化表中列出对虚  基类构造函数的调用,以初始化在虚基类中定义的数据成员。

(2)建立一个对象时,如果这个对象中含有从虚基类继承来的成员,则虚基类的成员是  由最远派生类的构造函数通过调用虚基类的构造函数进行初始化的。该派生类的其他基类  对虚基类构造函数的调用都自动被忽略。

(3)若同一层次中同时包含虚基类和非虚基类,应先调用虚基类的构造函数,再调用非  虚基类的构造函数,最后调用派生类构造函数。

(4)对于多个虚基类,构造函数的执行顺序仍然是先左后右,自上而下。

(5)对于非虚基类,构造函数的执行顺序仍是先左后右,自上而下。

(6)若虚基类由非虚基类派生而来,则仍然先调用基类构造函数,再调用派生类的构造  函数。