第十三章主要讲了类继承。

在一个类中派生出另一个类成为类继承，原始类称为基类，继承类称为派生类。派生类可以储存基类的数据成员，可以使用基类的方法，派生类也可以定义自己的成员和成员函数。派生类不能直接访问基类的私有成员，只能通过基类方法来进行访问。使用基类的指针可以在不进行显式类型转换的情况下指向派生类对象；基类的引用可以在不进行显式类型转换的情况下引用派生类的对象，然而基类的指针或引用只能用于调用基类的方法，不能调用派生类的方法。这种情况是单向的，即不可以讲基类对象和地址赋给派生类的引用和指针。在构造对象时，系统会自动调用隐式复制构造函数，从而实现把派生类中基类的对象复制给另一个基类，同理也能反过来进行。派生类和基类构建起一个派生类可以对基类对象执行任何操作，也可以对派生类对象执行操作的关系。继承只能在基类的基础上添加属性，但不能删除基类的属性。

多态指对于一个对象进行同名操作，会因为对象的类型不同而进行的操作不同。在程序中通过使用虚函数来实现。在使用引用或者时指针调用方法时，如果访问的函数不是虚函数，则用基类的引用或指针只能使用基类的函数，而不能根据对象选择函数；用关键字virtual后，可以通过指针指向的对象来选择方法，从而达到多态的实现。在基类中将派生类重新定义的方法声明为虚方法，方法在基类被定义为虚后，他在派生类中将自动成为虚方法，然而在派生类中使用virtual关键字可以方便查看。

虚函数的工作原理是给每个对象添加一个指向函数地址数组的指针，这数组称为虚函数表，虚函数表中存储了为类对象进行声明的虚函数的地址。无论类中包含多少个虚函数，在一个对象中仅需要添加一个指向虚函数表的指针，只是虚函数表的大小不同。使用虚函数会导致效率降低。构造函数不能为虚函数，而派生类的析构函数应该为虚函数。友元函数不能为虚函数，因为友元函数不是类成员。如果重新定义继承的方法，应该确保与原来的原型完全相同，如果返回类型是基类引用或指针，则可以修改为指向派生类的引用或指针。如果基类声明被重载了，则应该在派生类中重新定义所有基类版本。

在不涉及继承的情况下，关键字private和protected类似，当涉及继承时，对于外部世界来说，protected和private类似，对于派生类来说，protected和public类似。

当两个类并不是简单的主从关系，而是包含共有部分时，可以通过再设计一个基类来形容他们的共有部分，这个基类中，要包含纯虚函数来提供为实现的函数，定义方法为结尾处加=0。包含纯虚函数的类只能作为基类，