类继承可以从已有的类派生出新的类，派生类继承了原有类(基类)的特征。

从基类派生出新的类：

class NewclassName : public BaseclassName

{

…

}

冒号指出NewclassName类的基类是BaseclassName。BaseclassName是一个公有基类，这被称为公有派生。使用公有派生，基类的公有成员成为派生类的公有成员；基类的私有成员也将成为派生类的一部分，但只能通过基类的公有和保护方法访问。

派生类对象存储了基类的数据成员，可以使用基类的方法。派生类需要自己的构造函数，可以根据需要添加额外的数据成员和成员函数。构造函数必须给新成员和继承的成员提供数据。

派生类不能直接访问基类的私有成员，必须通过基类的方法访问，即派生类构造函数必须使用基类构造函数。如P484页的构造函数示例。

创建派生类对象时，程序首先调用基类构造函数，然后调用派生类构造函数。派生类构造函数总是调用一个基类构造函数。可以使用初始化器列表语法指明使用的基类构造函数，复制将使用默认的基类构造函数。派生类对象过期时，程序将首先调用派生类析构函数，然后调用基类析构函数。

派生类可以使用基类的方法，条件是方法不是私有的；基类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；基类引用可以在不进行显式类型转换的情况下引用派生类对象。但基类指针或引用只能调用基类方法，不能使用派生类的方法。不能讲基类对象和地址赋给派生类引用和指针。

基类引用定义的函数或指针参数可以用于基类对象或派生类对象(示例P488-489)。

多态。希望同一个方法在派生类和基类中的行为是不同的。2种实现多态的方法：1.在派生类中重新定义基类的方法；2.使用虚方法。

virtual关键字。

若没有使用了virtual，程序将根据引用类型或指针类型选择方法；若使用了virtual关键字，程序将根据引用或指针指向的对象的类型来选择方法。

通常在基类中将派生类会重新定义的方法声明为虚方法。方法在基类中被声明为虚的后，它在派生类中将自动称为虚方法。virtual只用于方法原型，不用于方法定义。P493例子。

**联编(binding)**：将源代码中的函数调用解释为执行特定的函数代码块称为函数名联编。在编译过程中完成联编，称为静态联编(static binding),或早期联编；但因为虚函数使得不能在编译时确定使用的函数，因此编译器在程序运行时选择正确的虚方法代码，称为动态联编(dynamic binding),或晚期联编。reP502

访问控制：protected

protected与private类似，只能用公有类成员访问。区别是派生类的成员可以访问基类的protected成员，但不能访问private成员。

抽象基类(abstract base class, ABC)。从Ellipse和Circle类中抽象出他们的共性，将这些特性放到一个ABC中，从ABC派生出Ellipse类和Circle类。

C++中使用纯虚函数(pure virtual function)提供未实现的函数，纯虚函数的结尾处为=0，如：

virtual double Area() const=0;

当类声明中包含纯虚函数时，不能创建该类的对象，因为包含纯虚函数的类只能用作基类，要称为真正的抽象基类ABC，必须至少包含一个纯虚函数。

示例P510

继承和动态内存分配

P516re