第五章

静态联编是指系统在编译时就决定如何实现某一动作。静态联编要求在程序编译时就知道调用函数的全部信息。因此,这种联编类型的函数调用速度很快。效率高是静态联编的主要优点。  
动态联编是指系统在运行时动态实现某一动作。采用这种联编方式,一直要到程序运行时才能确定调用哪个函数。动态联编的主要优点是:提供了更好的灵活性、问题抽象性和程序易维护性。  
静态联编支持的多态性称为编译时多态性,也称静态多态性。在C++中,编译时多态性是通过函数重载(包括运算符重载)和模板实现的。利用函数重载机制,在调用同名的函数时,编译系统可根据实参的具体情况确立所要调用的是哪个函数。

运算符重载：在C++中，除了可以对函数重载外,还可以对大多数运算符实施重载。运算符重载与函数重载相比,稍微复杂一些。运算符重载是通过创建运算符重载函数来实现的。运算符重载函数可以是在类外定义的普通函数,也可以是类的成员函数或友元函数。本节将分别介绍这三种运算符重载函数。  
系统预定义类型间的转换：类型转换是将一种类型的值转换为另一种类型值。对于系统预定义的标准类型(如int, float, double,char等),C++提供两种类型转换方式:一种是隐式类型转换;另一种是显式类型转换。  
虚函数：虚函数是重载的另一种表现形式。这是一种动态的重载方式,它提供了一种更为灵活的运行时的多态性机制。虚函数允许函数调用与函数体之间的联系在运行时才建立，也就是在运行时才决定如何动作,即所谓的动态联编。下面先介绍引入派生类后的对象指针，然后再介绍虚函数。