高等职业教育计算机技术专业贯通制教材

C++面向对象程序设计

主编　胡 云　　副主编　马海珠　卫洛斌

本书配有电子教学参考资料包

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等职业教育计算机技术专业贯通制教材

C++面向对象程序设计

主　编　胡　云
副主编　马海珠
　　卫洛斌

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书着眼于技能型紧缺人才培养目标，以面向对象的思想详细地介绍了运用C++语言进行程序设计和开发的知识。本书共分10章。主要内容包括：C++编程基础、数据类型和表达式、控制结构、函数、数组与指针、构造函数与析构函数、函数重载和运算符重载、继承、虚函数与多态性、C++流和项目实践。全书坚持以能力培养为导向和目标，突出实用性、适用性和先进性，结合案例深入浅出、循序渐进地引导读者学习。各章均配有习题和上机练习。

本书可作为高等职业院校计算机类专业的教材，也可作为C++程序开发维护人员的参考书。

图书在版编目（CIP）数据

C++面向对象程序设计/胡云主编. —北京：电子工业出版社，2008.8
高等职业教育计算机技术专业贯通制教材
ISBN 978-7-121-06788-4
Ⅰ. C… Ⅱ. 胡… Ⅲ. C语言－程序设计－高等学校：技术学校－教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第100925号

策划编辑：施玉新
责任编辑：施玉新 牛旭东 特约编辑：刘 皎
印 刷：
装 订：北京牛山世兴印刷厂
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编 100036
开 本：787×1 092 1/16 印张：13.75 字数：352千字
印 次：2008年8月第1次印刷
印 数：3 000册 定价：21.00元

凡所购买电子工业出版社图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系及邮购电话：（010）88254888。
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。
服务热线：（010）88258888。

前言

随着计算机技术的发展、用户软件需求的不断增长，以及软件规模的不断扩大，早期的C语言虽然具有功能丰富、语句简洁、语法灵活、数据结构多样化、能对硬件进行操作、高移植性和通用性等诸多优点，但用此进行软件的开发已经“力不从心”。C++语言正是在这种情况之下应运而生的。

C++语言是从C语言发展演变而来的，是C语言的超集。C++语言也是目前使用最广泛的一种高级程序设计语言，它既可以用于过程化程序设计，也可以用于面向对象的程序设计。它实现了类的封装、数据隐藏、继承及多态，使得其代码容易维护且高度可重用。

本书根据学生在接受、领会C++语言过程中的特点，注重通过大量生动实际的例子，深入浅出地讲解和分析复杂的概念，力图使读者通过实际动手领会和掌握C++的精髓，获得良好的学习效果，并能在实践中学以致用，获得良好的学习效果，为今后的进一步学习和应用开发打下坚实的基础。

本书共分10章，以面向对象的思想详细地介绍了运用C++语言进行程序设计和开发的知识。第1章介绍C++语言的发展史、面向对象的基本概念、C++语言中的基本概念、程序风格、运行环境等；第2章讲述各种常见的数据类型、各种表达式及其求值、表达式的优先级、结合性和副作用等；第3章介绍顺序结构、选择结构和循环结构三种基本结构及其相应的控制语句等；第4章介绍如何定义和使用函数等；第5章讨论如何使用数组与指针及其联系等；第6章阐述类中两个重要的成员函数：构造函数和析构函数；第7章对函数重载表达式重载的使用、友元函数和友元类等进行介绍；第8章主要介绍继承、虚函数和多态性等概念；第9章着重讨论流的分类及利用流对文件操作等；第10章综合前面所学知识和内容，系统介绍一个系统从需求分析、设计到最后实现的全部过程，使学生能尽快地将知识转化为能力。

在本书的编写过程中，作者力求体现职业教育的性质、任务和培养目标，坚持以能力培养为导向和目标，突出教材的实用性、适用性和先进性。本书从培养技能型紧缺人才的目的出发，采用案例驱动的教学方法，深入浅出、循序渐进地引导读者学习和掌握本课程的知识点。每章后面均附有习题和上机实验，可供读者自我测试之用。

本书第1章和第6章由时荣编写，第2章由马海珠编写，第3章由殷惠萍编写，第4章由黄玲娜编写，第5章由袁江琛编写，第7章和第8章由胡云编写，第9章和第10章由卫洛斌编写，全书由胡云统稿，江阴职业技术教育中心校王香菊主审。无锡高等师范学校的肖敏老师和无锡市广播电视大学现代教育中心的李盘荣老师在成书的过程中给予了多方面的指

导，提出了很好的意见和建议。此外无锡市广播电视大学信息工程系对本书的编写工作给予了有力的支持。在此，作者向所有对本书编写工作给予支持和帮助的人表示衷心的感谢。

在本书的编写过程中，我们参阅了大量的资料，在此对所有编著者表示衷心的感谢。由于编者水平有限，加之时间仓促，书中难免存在不足之处，敬请读者批评指正。

为了方便教师教学，本书还配有教学指南、电子教案及习题答案（电子版），请有此需要的教师登录华信教育资源网（www. huaxin. edu. cn 或 www. hxedu. com. cn）免费注册后再进行下载，在有问题时请在网站留言板留言或与电子工业出版社联系（E-mail：hxdu@ phei. com. cn）。

编　者

2008 年 4 月

目 录

第1章 C++编程基础

【本章学习要点】

1. 掌握面向对象的基本概念。
2. 掌握对象和类的概念，理解对象的特性。
3. 掌握C++语言的基本概念。

1.1 C++语言简介

1.1.1 C++语言的发展史

C++语言源于C语言，而C语言是在B语言的基础上发展起来的。1960年出现了一种面向问题的高级语言ALGOL60。1963年英国剑桥大学推出了CPL（Combined Programming Language）语言，后来经简化成为BCPL语言。1970年美国贝尔（Bell）实验室的K. Thompson以BCPL语言为基础，设计了一种类似于BCPL的语言，取其第一字母B，称为B语言。1972年美国贝尔实验室的Dennis M. Ritchie为克服B语言的诸多不足，在B语言的基础上重新设计了一种语言，取其第二字母C，故称为C语言。1980年贝尔实验室的Bjarne Stroustrup对C语言进行了扩充，多次修改后起名为C++。以后又经过不断的改进，发展成为今天的C++语言。

C++语言最初被称为“带类的C”。不仅保持了C语言的简洁、高效和可移植性好等特点，而且还支持面向对象的程序设计方法。尤其是在1998年推出了ANSI C++国际标准之后，C++语言的应用越来越广泛，其流行程度不亚于当初的C语言。

1.1.2 C++语言的特点

作为C语言的继承者，C++语言继承了C语言的特点：丰富的运算符和数据类型、结构化的程序设计方法、高效的机器代码、良好的可移植性。同时由于C++语言是C语言的超集，C++语言与C语言保持高效兼容，这使得许多C语言代码不需修改就可以在C++程序中使用。

C++语言扩展了C语言的功能，增加了面向对象机制。C++语言既支持传统的结构化程序设计，也支持当今的面向对象程序设计，利用面向对象程序设计技术实现了软件重用，提高了软件开发的效率。

C++语言既适用于编写系统软件，也适用于设计应用软件。作为程序设计语言，C++语

言的目标是为程序员的软件开发活动提供一个优良的设计工具，以编写模块化程度高、可重用性和可维护性俱佳的程序。可以说C++语言是程序员的语言。

与C语言相比，C++语言的错误检查机制强，它提供了专门的机制检查类，更适合大、中型程序的开发。同时，C++非常强调代码的有效性和紧凑性。事实表明，C++语言可用于C语言的所有应用领域，且其效果要比C语言好得多。

C++语言已经成为当今主流的程序设计语言，各大知名公司也都纷纷推出自己的C++编译器，如Microsoft公司的Visual C++、Borland公司的Borland C++、Inprise公司（已被Borland公司兼并）的C++ Builder和IBM公司的VisualAge C++等。

1.2 面向对象方法的基本概念

面向对象不只是一种程序设计方法，还是一种建立客观事物模型、分析复杂事物的思想方法，它是以人们通常描述现实世界的方法来描述要解决的问题。面向对象是目前成熟并流行的软件工程方法之一，主要包括面向对象分析和面向对象程序设计。面向对象程序设计是在吸取结构化程序设计（Structured Programming，SP）的优点基础上发展起来的一种新的程序设计方法，同时又在最大程度上解决了软件代码的重用和维护问题。

面向对象是C++中的主要概念，在学习C++之前首先要了解这些概念。

1.2.1 对象

按照面向对象的观点，对象（object）是现实世界中各种各样实际存在的事物，包括有形的对象和无形的对象。例如：人、学生、猫、动物、植物、汽车、工厂和计算机等都是有形的对象，而计划、思想、控制系统、程序等都是无形的对象。对象是构成世界的一个独立单位，它具有自己特定的属性（attribute）（如大小、形状和重量等）和行为（behavior）（如生长、行走、转弯和运算等），人们通过对象的属性和行为来认识对象。在计算机科学中，对象是系统中用来描述客观事物的一个实体，它是构成系统的基本单位，而系统可以看做是由一系列相互作用的对象组成的。

1.2.2 类

为了对具有相同属性（又称状态）和行为（又称操作、方法）的对象进行分类描述，引入了类（class）的概念。对对象进行分类的原则是抽象，即忽略对象的非本质特征，只考虑与当前求解问题有关的本质特征。类定义了同类对象的公共属性和行为，属性用数据结构表示，行为用函数表示，类可以用如下公式表示：

类 = 数据结构 + 对数据进行操作的函数

类是面向对象语言必须提供的用户定义的数据类型，它将具有相同状态、操作和访问机制的多个对象抽象成为一个对象类。

类是对象集合的抽象，规定了这些对象的公共属性和行为，类与对象的关系如同汽车与具体的一辆车的关系。汽车都能跑，有四个轮子，所有的汽车组成了一个类，具体到一辆汽车，它具有类的全部特性（能跑，有四个轮子），是汽车类的一个元素。类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。可见，一个对象又称为类的一个实例（instance）。

1.2.3 消息

面向对象方法提供了对象之间的通信机制。程序由一些相互作用的对象（类）构成，就像人们彼此之间互通信息一样，对象之间的交互通过发送消息来实现。程序通过执行对象的各种行为方法，来改变对象的状态（属性数据），从而使该对象发生某些事件。当对象发生某些事件时，通常需向其他相关对象发送消息，请求它们做出一些处理。

消息是向某对象请求服务的一种表达方法。对象内有方法和数据，外部的用户或对象对该对象提出了服务请求，可以称为向该对象发送了消息。

1.2.4 继承

客观世界的事物有很多相似性，其一个基本原因是事物之间具有某种“继承”关系。例如：人与古猿有许多相似之处，因为人是古猿进化而来。面向对象方法正是利用了客观世界的继承特性来构造对象，很好地解决了软件的可重用性问题。

继承（inheritance）是面向对象方法的一个特征，指一个新类可以从已有的类派生而来。新类继承了原有类的特性（即属性和行为）。并且，为了满足具体的需要，新类可以对原有类的行为进行修改，还可以增加新的属性和行为。例如：所有的 Windows 应用程序都有一个窗口，它们可以看做都是从一个窗口类派生出来的，但有的程序用于文字处理，有的程序用于表格操作，有的用于画图，这是根据具体需要，通过继承派生出了不同的类。类与类之间可以组成继承层次，一个类（子类）可以定义在另一个已定义类（父类）的基础上。子类不仅可以继承父类中的属性和操作，而且可以定义自己的属性和操作。

1.2.5 封装

封装（encapsulation）是面向对象的另一个特征，是对象和类的主要特性。它有两个含义：第一个含义是，把对象的全部属性和全部服务结合在一起，形成一个不可分割的独立单位（即对象）。第二个含义也称为“信息隐蔽”，即尽可能隐蔽对象的内部细节，对外形成一个边界（或者说形成一道屏障），只保留有限的对外接口使之与外部发生联系。这主要是指对象的外部不能直接地存取对象的属性，只能通过几个允许外部使用的服务与对象发生联系。封装保证了模块具有较好的独立性，使得程序维护修改较为容易。由于对应用程序的修改仅限于类的内部，因而可以将应用程序修改带来的影响减小到最低限度。

1.3 C++ 语言的基本概念

1.3.1 程序

为了对 C++ 程序结构有一个初步了解，下面通过一个简单例子来了解 C++ 程序。

【例 1.1】一个简单的 C++ 程序。

```
#include < iostream. h >
void main( void)
{
    / * 输出 This is my first C++ program. * /
```

```
    cout <<" This is my first C++ program. \n" ;
}
```

这个程序运行后在屏幕上输出字符串“This is my first C++ program.”。我们可以发现：

（1）C++源程序文件的扩展名为.cpp。

（2）C++注释不但可以使用符号“/*”和“*/”，表示符号“/*”和“*/”之间的内容都是注释；而且还可以使用一个双斜线“//”，表示“//”之后的本行内容是注释，注释在按回车键后自动结束。

（3）C++程序一般包含的是标准输入、输出流的头文件iostream.h，输入、输出可以通过使用输入、输出流对象（如cin、cout）来完成。

一个结构化的C++程序可以由多个函数构成，函数与函数之间是相对独立的，它们的定义是并行的，一个函数可以被其他函数调用。每个程序都从主函数main()开始执行，从主函数返回时结束执行。如同其他高级编程语言一样，C++程序可以由多个源文件构成，它们分别被编译成目标代码，然后连接成一个可执行程序。

1.3.2 对象和类

从面向对象的角度来说，类是对某一类对象的抽象，而对象是类的具体实例；从程序设计语言的角度来说，类是一种复杂的自定义数据类型，对象是属于这种数据类型的变量。C++引入了类这种抽象数据类型，实现了对对象的抽象和封装。

C++类的结构比较复杂，可以将其看做是一种既包含数据又包含函数的数据类型。显然，描述不同编程对象的类应有不同的内部结构，在使用类来声明对象前应先定义其结构。C++定义类的基本形式如下：

```
class <类名>
{
private:
    <私有数据成员和私有成员函数的声明列表>;
public:
    <公有数据成员和公有成员函数的声明列表>;
protected:
    <保护数据成员和保护成员函数的声明列表>;
};
```

其中，class是定义类的关键字，<类名>是用户自定义的标识符，花括号括起来的部分称为类体，它包括所有数据成员和成员函数的声明。数据成员又称成员变量（member variable），成员函数又称为方法（method）。关键字private、public和protected称为访问权限控制符，用来设置数据成员和成员函数的访问权限属性。

private属性表示数据成员或成员函数是类的私有成员，这类成员只允许被本类的成员函数访问或调用；public属性表示数据成员或成员函数是公有成员，这类成员允许被本类或其他类的成员函数（通过对象）访问或调用；protected属性表示数据成员或成员函数是保护成员，这类成员允许被本类的成员函数和派生类的成员函数访问或调用。

在类的外部通过对象只能访问所属类的公有成员，而私有成员只能在类的成员函数中被访问。一般而言，数据成员定义为private属性；成员函数定义为public属性，作为类的外

部接口。在声明成员时如果省略了访问权限控制符，则其属性默认为 private。

【例 1.2】定义类 Time（时间）。

```
class Time{
private:
    int hour;
    int minute;
    int second;
public:
    void setTime(int,int,int);
    void showTime();
};
```

类 Time 有 hour、minute 和 second 三个私有数据成员，它们只能在类的成员函数中被访问或赋值；类 Time 有两个公有成员函数，可在类的外部被调用，它们可视为访问类 Time 的外部接口，但必须通过一个对象实现对其调用。

利用 C++ 类进行面向对象程序设计，以上声明类的成员只是完成了任务的第一步；最重要的任务是实现定义的类。类的实现实质上是类的成员函数的实现，即定义类的成员函数。成员函数的定义形式与一般函数的定义基本相同，但如果在类的外部定义成员函数，必须在成员函数名前加上类名和作用域限定符 ::。

【例 1.3】类 Time 的实现。

```
void Time::setTime(int h,int m,int s)
{
    hour = (h>=0&&h<24)?h:0;                    //设置时间
    minute = (m>=0&&m<60)?m:0;
    second = (s>=0&&s<60)?s:0;
}
void Time::showTime()
{
    cout<<hour<<':'<<minute<<':'<<second<<endl;          //输出时间
}
```

由于不允许在类的外部访问或修改类 Time 的私有数据成员 hour、minute 和 second，所以类 Time 增加两个公有成员函数 setTime() 和 showTime()，以供在类的外部设置或显示类 Time 的私有数据成员 hour、minute 和 second。

一般将类的定义放在头文件（*.h）中，类的实现放在源文件（*.cpp）中，而 main 主函数可以放在另一个源文件中。在源文件中用#include 编译预处理指令包含头文件。

对象是类的一个实例，定义并实现了类后，就可以利用类来声明对象，其形式与普通变量的声明类似。例如，以下用类 Time 声明了对象 tl、today 和对象的指针（详见第 5 章）ptl：

```
Time tl,today;      //声明对象 tl、today
Time *ptl = &tl;    //声明指向对象 tl 的指针 ptl
```

声明对象后，就可以像引用结构变量一样，利用成员运算符“.”或指向运算符“->”

引用对象的公有成员，但注意不能引用对象的非公有成员。

【例1.4】类 Time 的使用，声明对象并设置对象属性。

```
void main( )
{
    Time EndTime;                    //声明对象 EndTime
    EndTime. setTime(1,2,3);         //设置对象 EndTime 的时间(属性,数据成员)
    cout <<" The end time is:" ;
    EndTime. showTime( );          //显示对象 EndTime 的时间
}
```

1.3.3 常量和变量

在程序中使用的数据有常量和变量两种类型，常量的值是始终不变的，而变量的值是可以被改变的。常量和变量的主要区别在于：常量不占内存空间，不能为常量赋值；而变量需要占内存空间，可以给变量赋不同的值。不管常量还是变量，程序中使用的每一个数据都属于一种特定的数据类型。

常量的书写方式规定了该常量的数据类型，而变量在使用之前必须先进行变量的声明(declaration)，以便编译程序为变量分配合适的内存空间，并可以给变量赋一个初值(即初始化)。变量声明语句的一般形式如下：

<数据类型><变量名1>[=<初始值1>],<变量名2>[=<初始值2>],…;

例如：

```
float x,y,z;                    //声明3个浮点型变量 x、y、z
int total,num =2008;            //声明整型变量 total、num，并对 num 初始化
```

变量声明语句定义了变量的名称和数据类型，在程序中通过变量名存取其中的数据，数据类型说明了变量所占用空间的大小和可以进行的运算，C++基本数据类型见2.1节。变量还具有作用域和存储类型，相关内容在第4章中进行详细介绍。

1.3.4 函数

函数是C++程序的构成基础。C++程序都是由一个个函数所组成的。一个C++程序无论多么复杂，规模有多么大，程序的设计最终都落实到一个个函数的设计与编写上。

在C++中，函数是结构化设计的“自顶向下、逐步求精”思想的具体体现。函数是程序模块划分的基本单位，程序员可将一个复杂的程序分解为若干个相对独立且功能单一的子程序即函数进行设计。

C++函数有三种：主函数（即main( )函数）、C++提供的库函数和自定义函数。C++系统函数库提供了几百个完成不同功能的函数，编程时可以直接拿来使用。合理地编写用户自定义函数，可以简化程序模块的结构，便于阅读和调试，是结构化程序设计方法的主要思想之一。

1.3.5 输入和输出

程序从外部设备获得数据称为输入（input），反之，将程序中的数据送到外部设备，如屏幕、打印机等，称为程序的输出（output）。

大多数 C++ 程序都要有数据的输入/输出。在此，先简单地介绍 C++ 程序中最常用的标准输入/输出语句。

1. 标准输出语句

```
cout <<" let\'s learn to write a C++ Program. " ;
cout << endl;
cout <<" chicken hen cock"<< endl;
cout <<" "<< chicken <<" "<< hen <<" "<< cock << endl;
```

上面是四条标准输出语句。cout 是标准输出文件（默认为屏幕）的名字，“ << ”称为插入运算符。它们完成向显示器屏幕输出指定的内容。

第一条语句输出一个字符串常量，即在屏幕上显示包括空格和标点符号在内的一个字符串；

第二条语句中 endl 有特定的含义，它表示“回车换行”，即下一次输出数据从下一行左端开始。

第三条语句则连续向屏幕输出两项内容，先是字符串常量“chicken hen cock”，然后换行。它等价于下面两条语句：

```
cout <<" chicken hen cock" ;
cout << endl;
```

第四条语句等价于七条输出语句：

```
cout <<" " ;
cout << chicken;
cout <<" " ;
cout << hen;
cout <<" " ;
cout << cock;
cout << endl;
```

其中，chicken、hen、cock 不是字符串（它们没有引号），而是变量，输出变量当前的值。

2. 标准输入语句

下面的程序使用了标准输入语句 cin >> myage;

```
#include < iostream. h >
void main( void)
{
    int myage;
    cout <<" My age is" ;
    cin >> myage;                    //输入年龄(一个整数)
    cout << myage << endl;
}
```

这时 C++ 程序允许从键盘输入一个整数，例如 21，结果在屏幕上显示：

My age is 21

cin 与 cout 类似，是标准输入文件（指键盘）的名字。“ >> ”称为提取运算符。myage 是一个程序员定义的整型变量名，变量不同于常量，在程序运行中其值可以改变。该语句的作用是把从键盘输入的值赋给变量 myage。

1.3.6 预处理命令#include

编译预处理是指在对源程序进行编译之前，编译预处理程序（precompiler）根据源程序中的编译预处理指令对源程序预处理。编译预处理程序是编译器的一部分，由编译器自动启动。与一般的 C++ 语句不同的是，编译预处理指令以符号“#”开头，结尾不使用分号“;”。编译预处理指令改善了编程环境，提高了编程效率。

C++ 提供的预处理有宏定义命令、文件包含命令和条件编译命令三种。我们着重介绍#include文件包含指令。

#include 文件包含指令是指将一个源文件嵌入到当前源文件中该指令处。#include 指令有以下两种形式，其格式如下：

#include < 文件名 >

#include" 文件名"

例如：

```
#include < stdlib. h >                    //stdlib. h:声明公共的系统标准函数
#include" MyPrg. h"                       //MyPrg. h:声明用户自定义的常量、变量及函数
```

第一种形式中，所要嵌入源文件用尖括号括起来。这种形式的#include 指令告诉编译预处理程序在编译器自带的外部库的头文件中搜索要嵌入的文件，它们一般是系统提供的公共头文件，存放在系统目录中的 Include 子目录下。

第二种形式中，所要嵌入的源文件用双引号括起来。这种形式的#include 指令告诉编译预处理程序先在当前子目录搜索要嵌入的文件（一般是用户自定义的头文件或源文件），如果没有找到文件，则再去搜索编译器自带的或外部库的头文件。

按照 C++ 函数使用要求，如果函数调用在前、函数定义在后，或者调用其他文件中（如系统库）定义的函数时，必须先进行函数声明。系统函数按其功能被分成几个库，对应每个库都有一个头文件，头文件的内容是某一类函数的原型声明。显然，只需在程序中使用#include 指令包含相应的头文件，而不必在程序中直接给出函数的声明。

以多文件方式组织的程序常常需要在各文件之间共享一些常量声明、变量声明、结构声明、函数声明和宏定义，可以将这些语句放在一个 C++ 头文件中（以 .h 作为扩展名），然后利用#include 指令将该头文件包含到需要它的源文件中。

1.3.7 头文件

每个 C++ 程序通常分为两个文件，一个文件用于保存程序的声明（declaration），称为头文件；另一个文件用于保存程序的实现（implementation），称为定义（definition）文件。

C++ 程序的头文件以“. h”为后缀。

头文件由三部分内容组成：

（1）头文件开头处的版权和版本声明；

（2）预处理块；