21世纪高职高专规划教材·计算机系列

C++程序设计及实训教程

肖 霞 编著

清华大学出版社
http://www.tup.tsinghua.edu.cn

北京交通大学出版社
http://press.bjtu.edu.cn

21世纪高职高专规划教材·计算机系列

C++程序设计及实训教程

肖　霞　编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

C++语言是近年来广泛使用的现代计算机语言，它既支持面向过程的程序设计，也支持基于对象和面向对象程序设计。许多高校均已陆续开设了C++程序设计课程。但是，由于C++语言涉及概念很多，语法比较复杂，内容十分广泛，许多人感到学习难度较大，难以入门。考虑到许多读者在学习C++语言前未学过其他语言，本书把入门起点降低到读者不需要具备C语言基础的程度。

本书内容全面、例题丰富、概念清晰、循序渐进且易于学习。主要内容有：绪论、数据类型及表达式、C++简单程序设计、数组、函数、指针和结构体、程序结构和编译预处理、类和对象、继承和派生、虚函数和多态性、输入/输出流、综合实训应用实例等。各章示例简易而典型，且备有丰富的编程练习和上机实训题，最后一章还给出了三个有代表性的较大的综合实训题目。配有光盘。

本书由教学一线的资深教师编写，适合作为各类高等院校C++语言程序设计课程的教材及该课程的实训教材，也可作为教师、学生或程序开发人员的参考书。

版权所有，翻印必究。举报电话：010－62782989　13501256678　13801310933
本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
本书防伪标签采用特殊防伪技术，用户可通过在图案表面涂抹清水，图案消失，水干后图案复现；或将表面膜揭下，放在白纸上用彩笔涂抹，图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

C++程序设计及实训教程／肖霞编著. —北京：清华大学出版社；北京交通大学出版社，2007.1
（21世纪高职高专规划教材·计算机系列）
ISBN 978－7－81082－880－2

Ⅰ.C…　Ⅱ.肖…　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（2006）第112341号

责任编辑：韩　乐　　特邀编辑：朱　宇
出版发行：清华大学出版社　　邮编：100084　　电话：010－62776969
　　北京交通大学出版社　　邮编：100044　　电话：010－51686414
印 刷 者：北京市梦宇印务有限公司
经　　销：全国新华书店
开　　本：185×260　　印张：17.5　　字数：448千字
版　　次：2007年1月第1版　　2007年1月第1次印刷
书　　号：ISBN 978－7－81082－880－2/TP·308
印　　数：1～4000册　　定价：26.00元

本书如有质量问题，请向北京交通大学出版社质监组反映。对您的意见和批评，我们表示欢迎和感谢。
投诉电话：010－51686043，51686008；传真：010－62225406；E-mail：press@center.bjtu.edu.cn。

21世纪高职高专规划教材·计算机系列
编审委员会成员名单

主任委员 李兰友　边奠英

副主任委员 周学毛　崔世钢　王学彬　丁桂芝　赵　伟
韩瑞功　汪志达

委　　员 （按姓名笔画排序）

马春荣	马　辉	万志平	万振凯	王一曙
王永平	王建明	尤晓玮	丰继林	尹绍宏
左文忠	叶　华	叶　伟	叶建波	付晓光
付慧生	冯平安	江　中	佟立本	刘　炜
刘建民	刘　晶	刘　颖	曲建民	孙培民
邢素萍	华铨平	吕新平	陈国震	陈小东
陈月波	陈跃安	李长明	李　可	李志奎
李　琳	李源生	李群明	李静东	邱希春
沈才梁	宋维堂	汪　繁	吴学毅	张文明
张宝忠	张家超	张　琦	金忠伟	林长春
林文信	罗春红	苗长云	竺士蒙	周智仁
孟德欣	柏万里	宫国顺	柳　炜	钮　静
胡敬佩	姚　策	赵英杰	高福成	贾建军
徐建俊	殷兆麟	唐　健	黄　斌	章春军
曹豫莪	程　琪	韩广峰	韩其睿	韩　劼
裘旭光	童爱红	谢　婷	曾瑶辉	管致锦
熊锡义	潘玫玫	薛永三	操静涛	鞠洪尧

出 版 说 明

高职高专教育是我国高等教育的重要组成部分，它的根本任务是培养生产、建设、管理和服务第一线需要的德、智、体、美全面发展的高等技术应用型专门人才，所培养的学生在掌握必要的基础理论和专业知识的基础上，应重点掌握从事本专业领域实际工作的基本知识和职业技能，因而与其对应的教材也必须有自己的体系和特色。

为了适应我国高职高专教育发展及其对教学改革和教材建设的需要，在教育部的指导下，我们在全国范围内组织并成立了“21世纪高职高专教育教材研究与编审委员会”（以下简称“教材研究与编审委员会”）。“教材研究与编审委员会”的成员单位皆为教学改革成效较大、办学特色鲜明、办学实力强的高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院，其中一些学校是国家重点建设的示范性职业技术学院。

为了保证规划教材的出版质量，“教材研究与编审委员会”在全国范围内选聘“21世纪高职高专规划教材编审委员会”（以下简称“教材编审委员会”）成员和征集教材，并要求“教材编审委员会”成员和规划教材的编著者必须是从事高职高专教学第一线的优秀教师或生产第一线的专家。“教材编审委员会”组织各专业的专家、教授对所征集的教材进行评选，对列选教材进行审定。

目前，“教材研究与编审委员会”计划用2～3年的时间出版各类高职高专教材200种，范围覆盖计算机应用、电子电气、财会与管理、商务英语等专业的主要课程。此次规划教材全部按教育部制定的“高职高专教育基础课程教学基本要求”编写，其中部分教材是教育部《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》的研究成果。此次规划教材编写按照突出应用性、实践性和针对性的原则编写并重组系列课程教材结构，力求反映高职高专课程和教学内容体系改革方向；反映当前教学的新内容，突出基础理论知识的应用和实践技能的培养；适应“实践的要求和岗位的需要”，不依照“学科”体系，即贴近岗位群，淡化学科；在兼顾理论和实践内容的同时，避免“全”而“深”的面面俱到，基础理论以应用为目的，以必需、够用为度；尽量体现新知识、新技术、新工艺、新方法，以利于学生综合素质的形成和科学思维方式与创新能力的培养。

此外，为了使规划教材更具广泛性、科学性、先进性和代表性，我们希望全国从事高职高专教育的院校能够积极加入到“教材研究与编审委员会”中来，推荐“教材编审委员会”成员和有特色、有创新的教材。同时，希望将教学实践中的意见与建议及时反馈给我们，以便对已出版的教材不断修订、完善，不断提高教材质量，完善教材体系，为社会奉献更多更新的与高职高专教育配套的高质量教材。

此次所有规划教材由全国重点大学出版社——清华大学出版社与北京交通大学出版社联合出版，适合于各类高等专科学校、高等职业学校、成人高等学校及高等院校主办的二级职业技术学院使用。

21世纪高职高专教育教材研究与编审委员会

2007年1月

此为试读，需要完整PDF请访问：www.ertongbook.com

前　言

本书针对现代教育教学改革理念，在提高教学效率的同时，有效地提高学生综合实践的能力。本书集结了多年来在软件开发与C++程序设计教学方面的实践，根据计算机专业教学改革特有的情况，探讨了现代计算机教学的规律，并基于实际应用进行编写，内容翔实，概念清晰。

本书的一个特点是入门起点较低，读者不需要具备C语言的基础。在学习C++语言前未学过其他语言的人也可以直接学习。本书在内容编排上注重基础性和实用性，也可以作为实践教学用书。

本书的另一个特点是通过实例引导读者尽快掌握C++程序设计的实际应用。每一章都配有实验内容，使读者通过对本章提供的实验内容进行练习，进一步熟练掌握知识点，并在书中最后通过三个大的实训实例帮助读者全面了解C++程序设计的方法和技巧，引导读者学习C++语言在实际编程中的应用。

本书分为12章。第1章～第7章对C++语言的基础内容（C语言）进行了讲解；第8章～第11章主要讲述了面向对象的技术，如类、对象的封装性，基类、派生类的继承性，重载、动态联编多态性等；第12章为C++语言学习后的集中实训或课程设计提供了实例，读者可以通过完善该部分内容进一步学好C++语言。

参加本书编写、审校的人员还有张海伟、李丹等教师，在此致以谢意。

由于作者水平有限，书中难免存在缺点和不足，恳请各位专家、老师和同学提出宝贵意见。本书中的所有程序代码均已调试通过，读者可以通过E-mail免费索取（作者E-mail地址：fzjsjxiao@163.com）。

编　者

2007年1月

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

第1章 绪 论

C++语言是一门优秀的程序设计语言，它比C语言更容易为人们学习和掌握，并且以其独特的语言机制在计算机科学领域中得到广泛的应用。C++语言充分支持面向对象程序设计，推进了面向对象技术的发展。

1.1 概述

1.1.1 面向对象方法的由来和发展

计算机科学发展的每一步都是在软件设计和程序设计语言中得到体现的。软件设计是一个发展的概念，随着软件开发规模的不断扩大和开发方式的变化，程序设计开始被人们作为一门科学对待，经过多年研究，在计算机科学中发展了许多程序设计方法。

下面我们通过回顾计算机语言的发展过程，了解一下面向对象的方法是如何产生的。

20世纪50年代的程序设计都是用机器语言或汇编语言编写的。这种程序的设计相当麻烦，严重影响了计算机的普及应用。随着计算机的应用日益广泛，发展了一系列不同风格的、为不同对象服务的程序设计语言。

最早的高级语言是在20世纪50年代中期研制的FORTRAN语言，它在计算机语言发展史上具有划时代意义。该语言引进了许多现在仍然使用的程序设计概念，例如变量、数组、分支、循环等。但在使用中也发现了一些不足，例如不同部分的相同变量名容易发生错误。50年代后期，高级语言ALGOL在程序段内部对变量实施隔离。ALGOL 60提出了块结构的思想，实际上也是一种初级的封装。60年代开发的Simula 67是面向对象的鼻祖。它将ALGOL 60块结构的概念向前推进了一步，提出了对象的概念。70年代出现的Ada语言是一种基于对象的语言，是支持数据抽象类型的最重要的语言之一，它丰富了面向对象的概念。到80年代中期以后，面向对象的程序设计语言广泛地应用于程序设计。

由于20世纪60年代末到70年代初出现了大型的软件系统，如操作系统、数据库等，因此给程序设计带来了新的问题。大型软件系统的研制需要花费大量的人力和物力，但编写出来的软件可靠性差、错误多且难以维护，已经到了程序员无法控制的地步，这就是“软件危机”。

1969年，E.W.Dijkstra首先提出了结构化程序设计的概念，他强调从程序结构和风格上研究程序设计，为“软件危机”起到了很大的缓解作用。70年代末，结构化设计方法将软件划分成若干个可单独命名和编址的部分，它们被称之为模块，模块化使软件能够有效地被管理和维护，能够有效地分解和处理复杂问题。

由于软件开发是对问题的求解过程，从认识论角度看，软件开发过程包括人们对要解决的问题及相关事物的认识和基于认识所进行的描述。而结构化设计方法不能直接反映出人类

认识问题的过程。随着计算机软件的发展，软件系统越来越复杂、庞大，结构化程序设计方法已显得力不从心。

20世纪80年代，面向对象的程序设计语言日趋成熟，在软件开发中各种概念和方法积累的基础上，就如何超越程序的复杂性障碍，如何在计算机系统中自然地表示客观世界的问题，人们提出了面向对象的设计方法，它不是以过程为中心，而是以对象代表的问题为中心环节，提出了“对象＋对象＋…＝程序设计”理论，使人们对复杂系统的认识过程与系统的程序设计实现过程尽可能地一致。面向对象的程序设计方法的出现，使“软件危机”得到很好的解决。

1.1.2 面向对象程序语言

20世纪80年代中期以后，面向对象的程序设计语言广泛地应用于程序设计，并且有许多新的发展。归纳起来大致可分两类：一类是纯面向对象的语言，如Smalltalk和Eiffel语言；另一类是混合型面向对象的语言，如C++语言和ObjectiveC语言。

C++语言是由AT&T公司贝尔实验室的Bjarne Stioustrup博士开发的，其创作灵感来源于计算机语言多方面成果的凝聚，特别是BCPL（Basic Combined Programming Language，C语言的基础）和Simula 67（以面向对象为核心的语言），同时也借鉴了ALGOL 68语言。C++的名字是由Rick Masenirti提出，到1983年确定。

C++语言是一门高效实用的混合型程序设计语言，它最初的设计目标是支持面向对象编程技术和抽象形态的类。C++语言包括两部分，一部分是C++基础部分，它是以C语言为核心的；另一部分是面向对象特性部分，是C++对C语言的扩充部分。这样它既支持面向对象程序设计方法，又支持结构化程序设计方法，同时广泛地应用基础和丰富的开发环境的支持，也使面向对象设计得到很快普及。

C++语言的基础部分与C语言相比，除了一些细微的差别外，可以说C++语言是C语言的加强版，它保留了C语言功能强、效率高、风格简洁、适合大多数系统程序设计任务等优点，使得C++语言与C语言之间取得了兼容性，因此，在过去的软件开发中积累的大量的C语言的库函数和实用程序都可在C++语言中应用。

另外，C++语言通过对C语言的扩充，克服了原有C语言的缺点，完全支持面向对象程序设计方法。它支持类的概念。类是一种封装数据和对这些数据进行操作的函数的用户定义的类型，使抽象得以描述。类还提供了数据隐蔽，确保了程序的稳定性、可靠性和可维护性。它还支持继承、派生和多态性等层次结构，使得其代码具有高度的可重用性。

面向对象程序设计的基本特点如下。

1. 抽象

抽象是人类认识问题最基本的手段之一。面向对象方法中的抽象是对具体问题（对象）进行概括，抽出一类对象的公共性质并加以描述的过程。事实上，对问题进行抽象的过程，就是一个分析问题、认识问题的过程。

2. 封装

抽象过程，就是认识和分析问题的过程，是程序设计的基础。面向对象程序设计的抽象，关键在于对问题本身进行整体分析，是将抽象得到的数据成员和代码成员相结合形成一个有机的整体，即数据与操作数据的行为的有机结合，这样的结合就是封装。在面向对象程

序设计中，通过对抽象结果进行封装，将一部分行为作为对外部的接口，而将数据和其他行为进行有效的隐藏，就可以做到对数据访问权限的合理控制。

利用封装的特性，编写程序时，对于已有的成果，使用者不必了解具体的实现细节，而只需要通过外部接口，依据特定的访问规则，就可以使用现有的东西。

3. 继承

分析问题，从抽象中得到数据和代码，并对其进行封装，得到有机的整体，应该说只是面向对象程序设计的初步工作。一个程序员，如果领会了抽象，问题就可以解决一半；如果懂得了封装，就能够起步进行程序设计。但是，任何问题都从头开始描述是不现实的。例如我们前面关于人类认识问题的讨论，人们总是处于一个不断深入的认识过程中，一个特定的问题，很可能前人已经进行过较为深入的探讨，这些结果如何利用？还有我们在程序设计的后期，对这个问题又有了深入的体会，这些新的认识成果如何加到已有的成果中？

继承，就是解决这个问题的良策。只有继承，才可以在别人认识的基础之上有所发展，有所突破，摆脱重复分析、重复开发的困境。C++语言中提供了类的继承机制，允许程序员在保持原有特性的基础上，进行更具体、更详细的类的定义。这样，通过类的这种层次结构，就可以反映出认识的发展过程。新的类由原有的类产生，因此我们说新类继承了原有类的特征，或者原有类派生出新类。关于继承和派生，将在以后的章节中进行详细的探讨。

4. 多态

如果说继承讨论的是类与类的层次关系，那么多态则是考虑这种层次关系及类自身成员函数之间的关系问题，是解决功能和行为的再抽象问题。多态，是指类中具有相似功能的不同函数使用同一个名称实现的现象。这也是人类思维方式的一种直接模拟，我们在日常生活中也常常有类似的用法，比如说“打球”这个“打”，就是一个多态现象，打篮球、打排球、打羽毛球，规则和实际“打”的操作相差甚远，只是功能相似，我们就统一使用“打”来表示，这实际上就是对多项运动的抽象。

1.1.3 Visual C++语言的发展

随着C++语言逐渐成为ANSI标准，这种新的面向对象设计语言迅速成为程序员最广泛使用的工具。众多C++语言的开发环境也随之不断推出，竞争十分激烈。1986年，Borland公司开发了Turbo C++程序语言，而后又推出了Borland C++版本。Microsoft公司于20世纪80年代中期在Microsoft C6.0的基础上开发了Microsoft C/C++7.0，同时引进了Microsoft Foundation Class（MFC）库1.0版本，完善了源代码。以前这些版本都是依赖于DOS环境，或在Windows下的DOS模式里运行。不久Microsoft公司推出了Microsoft C/C++8.0，即Visual C++1.0版本，它是Microsoft公司推出的第一个真正基于Windows环境下的可视化集成开发环境，将编辑、编译、链接和执行集成为一体。从Visual C++1.5版本以后，Microsoft公司决定不再将更多的努力花费在支持16位编程上，虽然Visual C++2.0仍提供对16位的支持，但从2.0版本以后，Visual C++更多地用来创建32位程序。在版本上，Microsoft公司没有推出3.0版本，版本号直接从2.0跳到4.0，这样Visual C++与MFC的版本号取得一致。由于Internet的流行明显地影响了产品的设计，在4.0版本中，Visual C++引进了为Internet编程而设计的新类库。5.0版本也增加了一些新类，但注意力更多地集中在改善产品的界面上，以提供一个更好的在线帮助系统，更高级的宏能力

和对开发者组内进行类与其他代码共享的支持。6.0 版本在功能上做了进一步的改进。Visual C++经历了 1.0 到 6.0 等版本的发展，软件系统逐渐庞大，功能日益完善。

1.2 C++程序的基本结构

了解和掌握程序的结构是编写程序的基础，就像是盖房子的地基一样重要。一般来说，一个 C++程序的基本框架结构包含了声明区、主函数和函数定义区三大部分。下面通过一个简单的例子认识 C++程序。

【例 1-1】 一个简单的 C++程序。

```
# include <iostream.h>
int add(int,int);
void main()
{
    int a,b;
    a=5;
    b=7;
    int c=add(a,b);
    cout<<"a+b="<<c<<endl;
}
int add(int x,int y)
{
     int z=x+y;
     return z;
}
```

程序运行结果为

```
a+b=12
```

1. 声明区

声明区出现在程序文件的所有函数的外部。但是，并不是每一个程序都有声明区，要视问题的不同而变化。它所包含的内容如下。

(1) 包含头文件。如例 1-1 中的#include<iostream.h>。

(2) 函数声明。如例 1-1 中的 int add(int, int)。

(3) 宏定义。例如#define PI 3.1415926。

(4) 全局变量声明。

(5) 类定义。例如 class name {…}。

(6) 结构体定义。例如 struct record {…}。

(7) 条件编译。例如# ifdeg…。

2. 主函数

主函数是 main 开始，是整个程序运行的入口，该函数中可能包含以下内容。

(1) 局部变量的声明。例如 int a,b。

(2) 函数调用。例如 int c=add(a,b)。

(3) 一般运算。例如 a=5。

(4) 结构控制。例如 if(a>b) c=a。

(5) 对象与结构的处理。

(6) 系统函数调用等。

3. 函数定义区

程序中除了主函数 main 外，还可以包含其他函数，每个函数都有一个不同的函数名称，以供主函数或其他函数调用。每个函数都是由函数说明和函数体组成的。如例 1-1 中的 add 函数。

1) 函数的说明部分

函数的说明部分包括函数返回值类型、函数名和函数的参数。

如例 1-1 中的 add 函数的说明部分为

```
int add(int x, int y)
```

函数有返回值是应指明返回值的类型；函数名后必须有一对圆括号"()"；函数的参数可以有也可以没有，没有参数的函数称为无参函数，有参数的函数称为有参函数，有多个参数时，参数之间用逗号隔开。

2) 函数体部分

函数体是用一对花括号"{}"括起来的用于完成某种功能的语句的集合。函数体一般包括变量定义和执行语句。在 C++ 语言中，一个变量必须在使用之前进行定义，变量的定义可出现在第一次使用之前的任意位置。

例如，

```
int add(int x, int y)
{
    int z;
    z = x + y;
    return z;
}
```

每一个语句（包含变量定义）的最后必须有一个分号。分号是 C++ 语言中不可缺少的部分，它表示一条语句的结束。若语句后缺少分号，程序在编译时会报出错。

4. 注释

注释是程序员对程序语句所做的说明，是提高程序的可读性的一种手段。C++ 的注释可用符号"//"注释，从该符号开始到该行的结束；也可用符号"/* …… */"注释从"/*"开始到"*/"结束的任意多行。

1.3 程序的调试和运行

Visual C++ 6.0 是 Microsoft 公司在 1998 年推出的基于 Windows 9x 和 Windows NT 的优秀集成开发环境。该开发环境为用户提供了良好的可视化编程环境，程序员可以利用该开发环境轻松地使用 C++ 源代码编辑器、资源编辑器和使用内部调试器，并且可以创建项目

此为试读，需要完整PDF请访问：www.ertongbook.com