

计算机软件开发与程序设计系列丛书

C/C++ 程序设计大全

(美) H.M.Deitel 著
P.J.Deitel

薛万鹏 等译

机械工业出版社
西蒙与舒斯特国际出版公司

本书分为三大部分。第一部分从第 1 章至第 14 章,详细叙述了 C 程序设计语言;第二部分从第 15 章至第 21 章,讲述了 C++ 语言和面向对象的程序设计;第三部分从附录 A 至附录 E,提供了支持正文的参考资料。

本书用完整的程序和输出范例来演示所讲的概念。每一章以学习目标和本章要目开头,中间列出了常见的程序错误和良好的程序设计习惯,最后以小结、本章涉及的术语和习题结束。

本书侧重于软件工程的原则,强调用结构化程序设计方法编写清晰的程序。

本书可作为没有或很少程序设计经验的读者的入门与提高教程。

H.M.Deitel, P.J.Deitel: C HOW TO PROGRAM, SECOND EDITION

Authorized translation from the English Language edition Published by Prentice-Hall, Inc.

Copyright 1994 by Prentice-Hall, Inc.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签,无标签者不得销售。

版权所有,翻印必究。

本书版权登记号:图字:01-97-0506

图书在版编目(CIP)数据

C/C++ 程序设计大全/(美)狄泰(Deitel, H.M.), (美)狄泰(Deitel, P.J.)
著;薛万鹏等译.-北京:机械工业出版社,1997.8

(计算机软件开发与程序设计系列丛书)

书名原文:C How to Program

ISBN 7-111-05816-X

I .C… II .①狄… ②狄… ③薛… III .C 语言-程序设计 IV .TP312

中国版本图书馆 CIP 数据核字(97)第 11791 号

出版人:马九荣(北京市百万庄南街 1 号 邮政编码 100037)

责任编辑:蒋 克

北京市密云县印刷厂印刷·新华书店北京发行所发行

1997 年 8 月第 1 版第 1 次印刷

787mm×1092mm 1/16·53 印张·1295 千字

印数:0001-7000 册

定价:89.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

C 语言表达能力强，目标代码效率高，可移植性好，既具有高级语言的优点，又具有低级语言的特点，因此特别适合编写操作系统、网络软件、编译器等系统软件。C++ 是 C 语言的超集，不仅增强了 C 语言的功能，而且具有面向对象的特点。在硬件技术飞速发展的今天，人们对软件功能的要求也越来越高。利用面向对象的程序设计技术实现软件的重用是加速软件开发进程的根本途径。C++ 已经成为当今主流程序设计语言。

市面上介绍 C/C++ 语言的书是很多的，这些书的内容可以说是“大而全”。C/C++ 语言入门并不难，但是，正如许多人认为的那样，掌握程序设计语言最困难之处是用其灵活高效地开发实际软件系统，这需要大量的实践和学习，而 C/C++ 因其丰富的功能和复杂的特点更是如此。

与其它书不同，本书从软件工程的角度介绍并讨论了 C/C++ 语言，读者能在学习的同时为实际使用打下坚实的基础，初学者和有经验的程序员都会从中受到启发。原书作者有着丰富的软件开发经验。Harvey H. Deitel 教授是虚拟存储系统（VMS）的先驱研究者之一，如今这种系统已经广泛地应用于 UNIX、OS/2 和 Windows NT 等等的操作系统上；Paul J. Deitel 在网络数据管理、数据库查询翻译器、金融财务管理系统的开发方面具有丰富的经验。本书的许多内容都体现了作者对程序设计技术的理解，正如作者所言：“本书集我们 40 余年程序设计经验之精华”。

本书主要由薛万鹏、纪宁、韩磊、许文轩、梅开、谢立、薛鸢、李岩、沈长华翻译，参加翻译工作的还有濮玉民、赵晓蓉、颜先杰、任映梅、治中、瞿跃龙、苏泳民、史荣光、上官立新、单力、汪梓鸣、成文、魏莲方、马蔚、杨开开、段群慧、蒋星、文达、韩青云等，梁敏、沈维亮、李昕怡对本书进行了全面的审校。华章图文信息有限公司和华译工作室也为本书的翻译给予了大力的支持和帮助，在此深表感谢。

原书内容严谨，具有较强的理论性和实用性。译者力求反映原书的特点和风貌，但由于时间关系及水平所限，不当和疏漏之处在所难免，敬请广大读者批评指正。

译者

1997 年 4 月

前 言

欢迎进入 C 语言世界！本书是由父子两代人合著而成。年长者 HMD 编写、教授程序设计已 30 余年，他经验丰富、强调程序的清晰优美。年轻人 PJD 编写程序亦有十余年，他精力充沛、热衷于教学和程序设计，他侧重于程序的性能和结果。我们希望通过我们两人的合作，奉献给读者一本集知识性和可读性于一体的书。

C 语言通常只教授给有程序设计基础的人。许多教育家认为 C 语言的复杂性及其众多的难点使得该语言不适合作为程序设计的第一门教学语言，而这正是本书的目标。那么我们为什么还要编写这本书呢？

事实上，C 语言已成为工业界选用的系统实现语言。有理由相信，C++（具有面向对象特点的改进的 C 语言）将成为 90 年代中后期的主流语言。一直在大学教授 Pascal 语言已 13 年的 Harvey Deitel 强调清晰的结构化程序。Pascal 课程中所讲的许多内容是结构化程序的基本原则。我们以 HMD 在大学里的教学方式编写了这本教材。我们会指出疑点并说明有效解决的步骤。我们的经验是：要以学习 Pascal 语言同样的方式学习该教材。不过有一个值得注意的不同点，那就是学生们正为他们在学一种毕业后就能立即有用的语言所鼓舞，这提高了他们的学习热情，在想到 C 语言难学时，这种学习热情对克服困难大有帮助。

出版本书的目的是为那些没有或很少程序设计经验的学生提供相当于大学水准的入门教程，同时也是要对传统 C 教程所要求的理论和实践作严密的论述。为了达到这些目标，该书的内容比其它 C 语言教材丰富得多。在我们的教学中约有 1000 名学生使用了该教材，好几万名外国读者详细地学习了该书的第一版。

为了让读者能够解决生动有趣的现实问题，本书提供丰富的范例、练习和从许多领域中提炼出的专题。

本书侧重于软件工程的原则，强调用结构化程序设计方法编写清晰的程序。本书力求避免使用模糊的术语和语法规范。

本书用完整的程序和输出范例来演示所讲的概念。每一章以学习目标和本章要目开头，中间列出了常见的程序设计错误和良好的程序设计习惯，最后以小结和本章涉及的术语结束。每一章还配有习题。在 PC 格式化磁盘和 Macintosh 格式化磁盘中有教师手册，手册中包含了教程中的程序和每一章末的练习答案。本书的练习涵盖面广，既有简单的复习题，也有复杂的程序设计问题，还有大型项目的开发。为了提高本书对学生的价值，我们在练习中投入了大量的精力。本书中的程序已在 ANSI C 兼容的编译器上作了测试，所用的机器包括 Sun SPARC 工作站、Apple Macintosh（Think C）IBM PC（Turbo C++，Borland C++）及 DEC VAX/VMS（VAX C）。

1. 关于本书

这本书具有帮助学习的如下特点：
学习目标

每一章均以学习目标开头，它告诉学生应该达到什么标准，并能够在学完该章后测试自己是否达到了标准。

本章要目

每一章的要目有助于读者按顺序学习，还能帮助读者预知要学习的内容和合理安排学习进度。

节

每章均由重点明确的节组成。每一节在介绍完整的 C 程序的同时介绍了 C 语言的特征。每一个程序之后都有输出结果，输出结果证实了所介绍的内容。把输出结果和程序设计语句结合起来讲解是学习和巩固所讲概念的好方法。本书的程序是为反映 C 语言的种种特点而设计的，在认真阅读的同时还需要上机实践。

图解

本书具有丰富的图解。结构化流程图有助于理解控制结构和结构化程序设计。本书还用大量的图示来说明重要数据结构（如链表、队列、堆栈、二叉树）的建立和维护。

有用的设计要素

本书帮助学生把学习重点放在程序开发、测试和调试、性能及可移植性这四大要素上。为此专门介绍了良好的程序设计习惯、常见的程序设计错误、性能忠告、可移植性忠告及软件工程评述。

良好的程序设计习惯

本书强调了良好的程序设计习惯。它使学生关注编写更好程序的技术。这些习惯反映了我们 40 年程序设计经验中的精华。

常见的程序设计错误

学生们在学习一种语言（尤其是第一门语言）时会犯某些共性的错误。让他们注意到这些错误会使他们受益无穷。

性能忠告

对于第一门程序设计课程来说，我们认为编写清晰、易于理解的程序是最重要的，但学生们希望编写的程序运行速度最快、占用内存最少、敲键盘次数最少等等。学生们非常关心性能，希望知道怎样提高程序的性能。为此，我们给学生们提供了性能忠告。

可移植性忠告

软件开发是一件既复杂又费钱的事。软件开发公司必须经常开发为各种计算机和操作系统定制的软件版本，所以特别强调程序的可移植性，也就是开发无需修改就能够在许多不同的计算机系统上运行的软件。许多人把 C 说成是开发可移植软件最好的语言，还有一些人认为用 C 实现的应用程序自动具有可移植性。事情并非这么简单，编写可移植性的程序需要细致的设计并且会有许多陷阱。ANSI 文件本身就用 11 页纸的篇幅介绍了实现可移植性的潜在的困难。本书包含了大量的可移植性忠告。书中结合了我们在开发可移植性软件方面的经验、以及 ANSI 标准和两本优秀的书对可移植性的论述（见第 1 章结束时列出的参考文献 Ja89 和 Ra90）。

软件工程评述

这种设计要素是第二版的新内容。本书归纳了影响软件系统（特别是大型的软件系统）结构和设计的大量的因素。

小结

每章的小结部分的内容可帮助学生复习和巩固重要的概念。还按次序列出了每一章中的良好的程序设计习惯、常见的程序设计错误、性能忠告、可移植性忠告及软件工程评述。

术语

每一章中还按字母顺序列出了本章定义的术语，它也是为了让学生进一步巩固所学的概念。然后归纳了良好的程序设计习惯、常见的程序设计错误、性能忠告、可移植性忠告及软件工程评述。

自我测验练习

为便于自学，本书还配有自我测验练习及答案。它可帮助学生树立信心并尝试其后的练习。

练习

每一章配有大量的练习。这些练习涵盖面广，包括重要术语及概念的回忆、编写一条 C 语句、写一小段 C 函数、编写完整的 C 函数和程序、直至编写重大项目。教师可根据需要和各学期的教学计划加以选择，还可以用这些练习题来安排家庭作业、小测验及重要考试。在 IBM-PC 格式化磁盘和 Apple-Macintosh 格式化磁盘中有这些练习题的答案。

2. 本书概况

本书分为三大部分。第一部分从第 1 章~第 14 章，详细地叙说了 C 程序设计语言（包括结构化程序设计的内容）。第二部分从第 15 章~第 21 章，讲述了 C++ 语言和面向对象的程序设计。第三部分从附录 A~附录 E，提供了支持正文的参考资料。

第 1 章“基本概念”，讨论什么是计算机、它是如何工作的以及人们是怎样为计算机编写程序的。引入了结构化设计的概念，解释了为什么这种技术促成了程序设计方法的革命。叙说了从机器语言到汇编语言及高级语言的发展史。介绍了 C 语言的起源和 C 程序设计的环境。

第 2 章“C 语言程序设计入门”，简要地介绍了 C 程序的编写，详述了 C 语言中的判断及算术运算。学完这一章后，学生将理解怎样编写简单而完整的 C 程序。

第 3 章“结构化程序的开发”，它可能是本书最重要的一章。本章介绍了帮助解决问题的算法表示，解释了结构化程序设计在编写易于理解、调试、维护及更可能在第一次测试就能正确运行的程序中的重要性。介绍了结构化程序设计的基本控制结构，即顺序结构、选择结构（if 和 if/else）和循环结构（while）。解释了自顶向下、逐步求精的程序设计技术，这种技术对编写正确的结构化程序是很重要的。介绍流行的帮助设计程序的方法——结构化伪码。第 3 章介绍的方法不仅适用于 C 语言，而且也适用于其它任何结构化程序设计语言。本章有助于培养学生良好的程序设计习惯，从而为解决复杂程序设计任务打下坚实的基础。

第 4 章“程序控制”，提炼了结构化程序设计的表示方法并介绍了其它控制结构。详细地探讨了循环结构，对比了用计数器控制的循环和用标记控制的循环。介绍了 for 结构，它是实现用计数器控制的循环的一种方便的方法。还介绍了 switch 选择结构和 do/while 循环结构。最后还讨论了逻辑运算符。

第 5 章“函数”，讨论了程序模块的设计和构造，介绍的内容包括标准库函数、程序员定义的函数、递归和传值调用。本章介绍的技术对于开发正确的结构化程序（尤其是大型程

序和软件) 相当重要。本章所介绍的“细化”程序设计策略是解决复杂问题的有效的方法, 这种策略就是用函数把复杂的问题分解为简单而又相关的部分。学生们欣赏本书对随机数和模拟的介绍, 还欣赏本书对完美地使用了控制结构的掷骰子游戏的讨论。本章还介绍了递归结构, 列出了书本(包括练习) 中涉及到的 31 例递归范例。有些教材把递归结构放在后面介绍, 但我们认为这一内容最好逐步覆盖全书。练习中搜集了几个经典的递归问题, 如汉诺塔问题(Towers of Hanoi)。

第 6 章“数组”, 讨论了数组数据结构(即同一类型的相关数据项的集合)。本章列举了大量的一维数组和二维数组的范例, 阐明了结构化数据与控制结构对于正确地开发结构化程序同样重要。本章的范例涉及到对数组的各种常用操作方法、打印直方图、数据排序、把数组传递给函数以及统计数据的简单分析。本章的特点之一是仔细地介绍二分查找法, 这种方法是对线性查找方法的极大的改进。练习中有许多有趣而富有挑战性的问题, 包括排序方法的改进、航空预约系统的设计、龟图概念(在 LOGO 语言中很著名)和在人工智能领域广泛作为程序开发概念的骑士漫游和八皇后问题。

第 7 章“指针”, 讲述了 C 语言中功能最强大的特点之一——指针。详细地解释了指针的操作、转引用调用、指针表达式、指针的算术运算、指针与数组的关系、指针数组和指向函数的指针。练习中有经典的龟兔赛跑的模拟、洗牌和发牌的算法, 还包括专题“建立自己的计算机”。专题中解释了机器语言程序设计的表示方法, 让学生设计和实现一个允许编写和运行机器语言程序的计算机模拟器。这种特点对于那些希望理解计算机到底是如何工作的读者尤其有用。学生们喜欢这种专题, 他们经常能够补充一些增强功能(练习中建议了许多增强功能)。在第 12 章中, 另一个专题指导了读者建立一个编译器, 然后把编译器所生成的机器语言放在第 7 章中建立的机器语言模拟器上运行。

第 8 章“字符和字符串”, 探讨了处理非数值数据的基本问题。这一章完整地介绍了 C 函数库中的字符和字符串处理函数。所讨论的技术可广泛地用于建立字处理器、页面和排版软件、文本处理应用程序。本章搜集了 33 道与文本处理有关的有趣的练习题, 涉及的领域包括回文、打油诗、英文转换、产生表示给定电话号码的七字单词、文本对齐、用单词书写支票金额、产生莫尔斯码、公制转换和讨债信。本章最后一道练习是对学生的挑战, 它要求用计算机化的字典建立纵横字谜发生器。

第 9 章“格式化输入/输出”, 介绍了 printf 和 scanf 函数的强大的格式化能力。讨论了 printf 函数的格式化输出功能, 如把浮点数精确到指定的小数位、数值的列对齐、左对齐和右对齐、插入直接信息、强制打印出加号、打印前导零、用指数表示法输出结果、八进制和十六进制的使用、以及域宽和精度的控制。讨论了 printf 函数的转义序列, 如光标的移动、特殊字符的打印和系统响铃等等。讨论了 scanf 函数的所有的格式化输入能力, 包括读取十进制、八进制、十六进制、浮点数、字符和字符串数据。还讨论了通过扫描输入的数据从扫描集中读取匹配或不匹配的字符。本章的练习实际上测试了 C 的所有的格式化输入/输出能力。

第 10 章“结构、联合、位运算和枚举”, 介绍了各种重要的特点。结构类似于 Pascal 语言和其它语言中的记录, 它把各种类型的数据项组合在一起。第 11 章用结构来建立包含以记录格式组织信息的文件。第 12 章把结构和指针以及动态内存分配一起使用, 建立了链表、队列、堆栈和树等等的动态数据结构。联合用于使一块内存区能够被不同类型的数据在不同

时刻使用。枚举对定义有用的符号常量是很方便的，它能够使程序具有更好的可读性。C 语言的功能强大的位操作能力能够让程序员编写控制底层硬件的程序。程序中使用位操作能够处理位串、屏蔽（或取消屏蔽）某个位和更紧凑地存储信息。这种通常只在底层汇编语言中才有的能力可以让程序员编写系统软件（如操作系统和网络软件）。本章的特点之一是重写了以前的洗牌和发牌模拟程序，新的程序效率更高，从中可以看出算法质量的重要性。

第 11 章“文件处理”，讨论了处理顺序存取文本文件和随机存取文本文件的技术。先介绍了数据的层次结构，即从位、字节、字段、记录到文件这样的一种层次结构。然后简要地介绍了 C 语言把文件看作字节流的观点。本章用三个程序讨论了顺序存取文件，这些程序说明了怎样打开和关闭文件、怎样按顺序把数据存储在文件中以及怎样按顺序读取文件中的数据。讨论随机存取文件时介绍了四个程序，这些程序说明了怎样按顺序建立一个随机存取的文件、怎样随机地读取和写入文件数据、怎样按顺序读取随机存取文件中的数据。第四个程序是一个完整的金融事务处理程序，它综合了顺序和随机存取文件的许多技术。工业界的读者告诉我们说，通过本书的学习，他们已经能够编写在其机构内立即有用的文件处理程序。

第 12 章“数据结构”，讨论了用于建立动态数据结构的技术。本章以讨论自引用结构和动态内存分配开始，然后讨论了怎样建立和维护各种动态数据结构，包括链表、队列、堆栈和树。对于每一种类型的数据结构，我们都提供了完整的程序和输出结果。第 12 章有助于学生真正地掌握指针。本章包含了使用间接引用和二次间接引用（特别难懂的概念）的丰富的范例。指针学习的难点之一是学生难以想象数据结构以及结点的连接。为此，我们用图示的方法来说明链节及其建立的顺序。用二叉树范例研究指针和动态数据结构是再好不过了。范例程序建立了一个二叉树，并以先根次序、中根次序和后根次序遍历该二叉树。学生们特别欣赏以中根次序遍历并打印结点的排序值。本章有丰富的练习，其中的重点之一是专题内容“建立自己的编译器”。该专题练习指导学生开发把中缀表达式转换为后缀表达式的程序和计算后缀表达式的程序。然后，我们修改了后缀表达式的算法，并用该算法产生机器语言代码。编译器把它所产生的机器语言代码文件放在了一个文件中（用第 11 章中的方法），学生可以在第 7 章开发的软件模拟器上运行编译器所产生的机器语言代码。

第 13 章“预处理程序”，详细地讨论了预处理程序的指令。本章完整地介绍了预处理指令 `#include` 和 `#define`。`#include` 指令在程序编译之前把指定文件的拷贝包含到指令所在的位置，`#define` 指令用来建立符号常量和宏。本章解释了条件编译，它能够让程序员控制预处理指令的执行和对源代码的编译。还讨论了把其操作数转换为字符串的运算符 `#` 和连接两个记号的运算符 `##`，介绍了五个预定义的宏常量 `__LINE__`、`__FILE__`、`__DATE__`、`__TIME__` 和 `__STDC__`。最后讨论了头文件 `assert.h` 中的宏 `assert`，在测试、调试和校验程序时，宏 `assert` 是有价值的。

第 14 章“高级话题”，介绍了在通常的教程中不涉及到的某些高级话题。第 14.2 节介绍了怎样把来自于文件的输入重定向到程序中、怎样把来自程序的输出重定向到文件中、怎样把一个程序的输出重定向成另一个程序的输入（即建立管道）以及怎样把程序的输出追加到现有的文件中。第 14.3 节讨论了怎样开发使用变长参数列表的函数。第 14.4 节介绍了怎样把命令行参数传递给函数 `main` 以及怎样在程序中使用这些参数。第 14.5 节讨论了怎样编译由多个源文件组成的程序。第 14.6 节讨论了用函数 `atexit` 注册要在程序结束时执行的函数，还讨论了结束程序执行的函数 `exit`。第 14.7 节讨论了类型限定符 `const` 和 `volatile`。第

14.8 节介绍了怎样怎样用整数和浮点数后缀指定数值常量的类型。第 14.9 节解释了二进制文件和临时文件的用法。第 14.10 节介绍了怎样用信号处理库捕捉意外的事件。第 14.11 节讨论了用函数 `calloc` 和 `realloc` 建立和使用动态数组。

本书的第一版只用一章的篇幅介绍 C++ 和面向对象的程序设计。在这期间，许多大学已经决定在 C 语言课程中介绍 C++ 和面向对象的程序设计。为此，这一版把对 C++ 和面向对象的程序设计的介绍增加到了 7 章，丰富的内容、练习和实验可供一学期学习使用。

第 15 章“把 C++ 看作更好的 C”，介绍了 C++ 中非面向对象的特点。这些特点改进了常规的面向过程的程序编写方法。本章讨论了单行注释、输入/输出流、声明、建立新的数据类型、函数原型和类型检查、内联函数（可取代宏）、引用参数、`const` 限定符、动态内存分配、默认参数、单目作用域运算符以及函数模板。

第 16 章“类和数据抽象（第一部分）”，把数据抽象的知识放在这一章中介绍是非常合适的。C++ 被认为是用来实现抽象数据类型（ADT）的。最近几年，数据抽象已经成为入门教学语言 Pascal 中的主要话题。在编写本书时，我们起初想把这部分内容放在 C 中介绍，但最终还是决定把它放在 C++ 中介绍。第 16、17 和 18 章涉及到了对抽象数据的处理。第 16 章讨论的内容包括：用结构实现抽象数据类型、用 C++ 风格的类实现抽象数据类型、访问函数和工具函数的使用、用构造函数初始化对象、用析构函数撤销对象、用默认的逐个成员拷贝的方式给对象赋值以及软件的可重用性。

第 17 章“类和数据抽象（第二部分）”，继续介绍类和数据抽象。本章讨论的内容包括：`const` 对象、`const` 成员函数、复合类（把其它类作为成员的类）、友元函数和友元类（他们对类的私有成员有特别的访问权）、`this` 指针（使对象能够知道自己的地址）、动态内存分配、类的所有对象共享的静态类成员、流行数据类型的范例（数组、字符串和队列）、容器类、递取类以及模板类。模板类是 C++ 语言最近新增的内容，它让程序员抓住抽象数据类型（如堆栈、数组和队列）的精髓，然后用最少的代码为特定的类型建立的 ADT（如整数类型的堆栈、浮点数类型的堆栈、整数类型的队列等等），模板类因此也常常称为带参数的类型。

第 18 章“运算符重载”，运算符重载是 C++ 教程中最流行的话题之一。学生们非常欣赏这方面的内容，他们发现这一章的内容与第 16 和 17 章对抽象数据类型的讨论非常匹配。利用运算符重载，程序员能够告诉编译器怎样把现有的运算符用在新类型的对象上。C++ 已经知道怎样把运算符用在内部类型的对象上（如整数、浮点数和字符）。但是，假设我们建立了字符串类，那么加号用在它上面有什么含义呢？（许多程序员用加号表示连接字符串）。这一章要学习怎样重载加号实现下述目的：把重载后的加号用于表达式中的两个字符串时，编译器要能够产生把两个字符串连接起来的运算符函数调用。本章讨论了运算符重载的基本知识、运算符重载的限制、成员函数和非成员函数的重载、单目和双目运算符的重载以及类型之间的转换。本章研究了丰富的实例，包括数组类、字符串类、日期类、巨型整数类和复数类（最后两个实例在练习中提供源代码）。

第 19 章“继承”，继承面向对象语言的基本功能之一，本章探讨了这方面的内容。继承是软件重用的一种方式，它能够在吸收现有类功能的基础上快速地开发出新的类和给新类添加新的功能。本章讨论了基类、派生类、受保护成员、公有继承、受保护继承、私有继承、直接基类、间接基类、基类和派生类中的构造函数和析构函数的使用以及有关继承的软件工

程。本章比较了继承（即“是”关系）和复合（即“有”关系），还介绍了对象的“使用”关系和“知道”关系。本章有丰富的实例研究，特别是用很长的篇幅实现了点、圆和圆柱体的类层次结构。本章还包含有关多重继承的实例，多重继承是 C++ 的高级特点。利用多重继承，派生类可以继承多个基类的属性和行为。

第 20 章“虚函数和多态性”，讨论了面向对象程序设计的另一种基本功能，即多态行为。当许多类因为继承共同基类而相关时，每一个派生类的对象都可以当作基类的对象处理，这能够使程序的编写不依赖于派生类对象的具体类型。同一个程序能够处理新类型的对象使得系统具有更好的可扩展性。多态性能够用更简单“直行”逻辑取代复杂的 switch 逻辑。例如，视频游戏的屏幕管理程序在绘制对象时只要简单地给链表中的每一个对象发送一条绘制消息就可以了。只要新类型的对象知道怎样绘制其自身，那么无需修改程序就可以把新的对象添加到程序中。当今流行的大量图形用户界面正是使用了这种程序设计风格。本章讨论了通过虚函数实现多态行为的机制，区分了抽象基类和具体类的差别（不用能抽象基类建立实例对象，而具体类可用来建立实例对象）。抽象基类可用于给派生提供可继承的接口。本章研究了两个重要的多态性实例，即工资单系统和在第 19 章中讨论过的点、圆和圆柱体这样一个类层次结构。

第 21 章“C++ 输入/输出流”，详细介绍了 C++ 中面向对象风格的输入/输出处理。许多 C 教程是基于 C++ 编译器教学的，教员通常更喜欢给学生们讲授新的 C++ 风格的 I/O（而不是 C 风格的 printf/scanf 函数）。本章讨论了 C++ 的各种输入/输出能力，包括用流插入运算符输出数据、用流提取运算符输入数据、类型安全的 I/O（是对 C 的改进）、格式化 I/O、无格式 I/O（用于提高性能）、用流操纵算子控制基数（十进制、八进制和十六进制）、浮点数、域宽控制、用户自定义算子、流格式状态、流错误状态、用户自定义类型的对象的状态以及把输出流系到输入流上（保证在要求用户通过键盘响应前先显示出提示信息）。

附录部分提供了有价值的参考资料。附录 A 是 C 语法概要，附录 B 摘编了 C 标准库，附录 C 完整地列出了运算符的优先级和结合性，附录 D 列出了 ASCII 字符集代码，附录 E 讨论了二进制、八进制、十进制和十六进制数值系统。附录 B 是根据 ANSI 标准文献摘编的（经美国国家标准化协会允许），该附录对编写 C 程序是详细而有价值的。附录 E 是一套完整的数值系统教程，其中包括自我测验练习和答案。

3. 致谢

我们要感谢许多为该书的编写默默奉献的人们。虽然本书的封面中没有列出他们的名字，但是没有他们的劳动、合作、热情和理解，本书的出版是不可能的。

HMD 要感谢他的诺瓦大学的同事们，他们是 Ed Simco、Clovis Tondo、Ed Lieblein Phil Adams、Raisa Szabo、Raul Salazar 和 Barbara Edge。

我们要感谢各界的朋友们，他们是数字设备公司的 Stephanie Stosur Schwartz、Sue - Lane Garrett、Janet Hebert、Faye Napet、Betsy Mills、Jennie Connolly、Barbara Couturier 和 Paul Sandore；太阳微公司的 Gary Morin；开放系统国际公司的 Bil Horst、David Litwack、Steve Hudson 和 Linc Faurer；信息服务公司的 Don Hall；信号培训公司的 Clive Lee；剑桥技术股份公司的 Gart Davis、Paul Sherman 和 Wilberto Martinez 以及把本书作为教材的用户们。

我们有幸与具有远见卓识的 Prentice Hall 出版商合作。在编辑出版方面，Joe Scordato 做了大量的工作。Dolore Mar 不仅完成了手稿的校对工作，而且总是在需要的时候给予真诚的帮助，她那饱满的热情令人敬佩。

主编 Marcia Horton 先生对本书的出版给予了极大的鼓励和支持。主管担负起编辑责任使人们对 Prentice Hall 充满了信任。我们高兴的是 Marcia 在管理工作日益繁忙之中仍与我们紧密地合作，她给我们留下了深刻的印象。

我们还要感谢第一版和第二版的审阅者。他们仔细地审阅了全书，为本书的精确性和完整性提出了大量的建议。他们是（以阅稿先后排列）：

David Falconer（加州州立大学）

David Finkel（伍斯特综合技术大学）

H.E. Dunsmore（普杜大学）

Jim Schmolze（图夫特大学）

Gene Spafford（普杜大学）

Clovis Tondo（IBM 公司和诺瓦大学的客座教授）

Jeffrey Esakov（宾夕法尼亚大学）

Tom Slezak（加州大学劳伦斯·利弗莫尔国家实验室）

Gary A. Wilson（威尔逊协会及加州大学百克利分校）

Mike Kogan（IBM 公司，32 位操作系统 OS/2 2.0 的主要设计者）

Don Kostuch（IBM 退休人员、世界范围内 C、C++ 和面向对象程序设计的指导者）

Ed Lieblein（诺瓦大学）

John Carroll（圣迭亚戈州立大学）

Alan Filipski（亚利桑那州立大学）

Greg Hidley（加州欧文大学）

Daniel Hirschberg（加州欧文大学）

Jack Tan（休斯敦大学）

Richard Alpert（波士顿大学）

Eric Bloom（本特来学院）

我们要特别感谢伦敦 QMW 大学计算机系的 Graem Ringwood 博士。他在该书教学过程中不断给我们提出富有建设性的建议，这些建议已经成为第二版中的重要内容。

Tem Nieto 先生在第 12 章的“建立自己的编译器”部分花费了大量的精力。

我们还要感谢许多教授、教员、学生以及其他专业人员对第一版所给予的指正（名单从略）。

这里，作者还要特别感谢软件工程领域的世界知名人物之一 Ed Lieblein 博士。他审阅了 C++ 和面向对象程序设计的内容。他是 HMD 在诺瓦大学的同事和朋友，是该大学计算机系的全职教授。此前他是开发编译器的牵头机构 Tartan 实验室的技术主任。在此之前，他是国防部长办公厅计算机软件和系统处主任。在他任责期间，他主管了国防部软件计划的制定，该计划专门用于改进未来关键系统中的软件能力。他提出了五角大楼 STARS 计划中的软件技术和软件重用性计划，指导了 Ada 程序的国际标准化。它还为在 Carnegie Mellon 大学建立软件工程协会起了重要的作用。和 Lieblein 博士在 Nova 大学的合作确实是我们的荣幸。

我们还要特别感谢 IBM 公司和诺瓦大学客座教授 Tondo 博士。Tondo 博士是本书审稿组的组长。他细致全面地审阅了全书，补充了不少 C 和 C++ 知识的细节。他还是《The C Answer Book》一书的作者之一，这本书是 Brian Kernighan 和 Dennis Ritchie 所著的经典著作《The C Programming Language》一书的练习答案。这两本书已经被广泛地使用。

本书是根据美国国家标准化协会 (ANSI) 和国际标准组织 (ISO) 颁布的 C 标准版编写的。征得美国国家标准化协会的同意，我们还广泛地使用了 ANSI 标准文献中的资料。我们诚挚地感谢 ANSI 出版社主任 Mary Clare Lynch 先生，他帮我们获得了必需的出版许可证。图 5-6、8-1、8-5、8-12、8-17、8-20、8-22、8-30、8-36、9-1、9-3、9-6、9-9、9-16、10-7 和 11-6 以及附录 A：《C 语法》和附录 B：《标准库》摘自 ANSI/ISO 9899: 1990。经许可，这些图作了压缩和修改。标准文献的拷贝可向美国国家标准化协会订购，地址是 11 W 42nd Street, New York, NY 10036。

最后，我们还要感谢 Barbara 和 Abbey Deitel，感谢她们的爱和理解以及在准备手稿方面所给予的帮助。他们测试了本书中的每一程序和校对了本书的手稿。他们敏锐的眼力从不放过手稿中每一个细微的错误。

本书中未发现的一切错误全部由我们承担。我们期待您的意见、批评、指正和建议。把您良好的建议提供给我们，衷心地感谢您为本书下一版所作出的贡献。我们的电子邮件地址是：

deitel@world.std.com

或按如下地址给我们写信：

Harvey M. Deitel

Paul J. Deitel

c/o Computer Science Editor

Prentice Hall

Englewood Cliffs, New Jersey 07632

我们会立即给予答复！

Harvey M. Deitel

Paul J. Deitel

目 录

译者序	
前言	
第 1 章 基本概念	1
1.1 引言	1
1.2 计算机是什么	3
1.3 计算机的结构	3
1.4 批处理、多道程序设计和分时	4
1.5 个人计算、分布式计算和客户/服务器 计算	4
1.6 机器语言、汇编语言和高级语言	4
1.7 C 语言的历史	5
1.8 C 标准库	6
1.9 其它高级语言	7
1.10 结构化程序设计	7
1.11 C 环境的基本知识	8
1.12 对 C 语言和本书的总的说明	9
1.13 ConcurrentC	10
1.14 面向对象的程序设计和 C++ 语言	10
第 2 章 C 语言程序设计入门	19
2.1 引言	19
2.2 一个简单的 C 语言程序：打印一行 文本	19
2.3 另一个简单的 C 语言程序：求两个 整数的和	23
2.4 内存的概念	27
2.5 C 语言的算术运算符	28
2.6 判断语句：相等测试运算符和关系 运算符	31
第 3 章 结构化程序的开发	48
3.1 引言	48
3.2 算法	48
3.3 伪码	49
3.4 控制结构	49
3.5 if 选择结构	51
3.6 if/else 选择结构	52
3.7 while 循环结构	56
3.8 制定算法：实例研究 1（计数器控制 的循环）	58
3.9 用自顶向下、逐步求精的方法制定算法： 实例研究 2（标记控制的循环）	59
3.10 用自顶向下、逐步求精的方法制定算法： 实例研究 3（嵌套控制结构）	65
3.11 赋值运算符	68
3.12 自增和自减运算符	69
第 4 章 程序控制	90
4.1 引言	90
4.2 循环的本质	90
4.3 计数器控制的循环	91
4.4 for 循环结构	93
4.5 for 结构：说明和评述	95
4.6 for 结构用法举例	96
4.7 switch 多路选择结构	99
4.8 do/while 循环结构	104
4.9 break 和 continue 语句	106
4.10 逻辑运算符	108
4.11 容易混淆的相等测试运算符（==） 和赋值运算符（=）	110
4.12 结构化程序设计小结	112
第 5 章 函数	129
5.1 引言	129
5.2 C 语言的程序模块	130
5.3 数学库函数	130
5.4 函数	132
5.5 函数定义	132
5.6 函数原型	137
5.7 头文件	139
5.8 函数调用：传值调用和传引用 调用	140
5.9 随机数的产生	140
5.10 范例：碰运气游戏	145
5.11 存储类别	147
5.12 作用域规则	150
5.13 递归	152

5.14 递归应用举例：斐波纳契 (Fibonacci) 数列	155	9.3 printf 的格式化输出	333
5.15 递归与迭代的比较	159	9.4 打印整数	333
第 6 章 数组	180	9.5 打印浮点数	335
6.1 引言	180	9.6 打印字符串和字符	336
6.2 数组	180	9.7 其它转换说明符	337
6.3 数组的声明	182	9.8 打印的域宽和精度	339
6.4 数组使用举例	182	9.9 在 printf 的格式控制串中使用 标志	341
6.5 把数组传递给函数	195	9.10 打印直接量和转义序列	344
6.6 数组排序	199	9.11 scanf 函数的格式化输入	345
6.7 实例研究：用数组计算均值 (mean)、 中位值 (median) 和众数 (mode)	201	第 10 章 结构、联合、位运算和 枚举	359
6.8 数组查找	205	10.1 引言	359
6.9 多维数组	210	10.2 结构的定义	359
第 7 章 指针	234	10.3 结构的初始化	361
7.1 引言	234	10.4 访问结构成员	362
7.2 指针变量的声明和初始化	234	10.5 结构和函数	363
7.3 指针运算符	235	10.6 类型定义：typedef	364
7.4 函数的传引用调用	238	10.7 范例：高效的洗牌和发牌模拟	365
7.5 对指针使用 const 限定符	241	10.8 联合	367
7.6 使用传引用调用的泡沫排序法	247	10.9 位运算符	370
7.7 指针表达式和指针的算术运算	251	10.10 位段	377
7.8 指针和数组的关系	254	10.11 枚举常量	381
7.9 指针数组	258	第 11 章 文件处理	393
7.10 实例研究：洗牌和发牌模拟	258	11.1 引言	393
7.11 指向函数的指针	263	11.2 数据的层次结构	393
专题：建立自己的编译器	278	11.3 文件和流	395
第 8 章 字符和字符串	290	11.4 建立顺序存取文件	396
8.1 引言	290	11.5 读取顺序存取文件中的数据	401
8.2 字符串和字符的基本知识	290	11.6 随机存取文件	405
8.3 字符处理函数库	291	11.7 建立随机存取文件	406
8.4 字符串转换函数	296	11.8 向随机存取文件中随机地写入 数据	408
8.5 标准输入/输出库函数	301	11.9 从随机存取文件中随机地读取 数据	410
8.6 字符串处理库中的字符操作函数	304	11.10 实例研究：事务处理程序	411
8.7 字符串处理库中的比较函数	306	第 12 章 数据结构	426
8.8 字符串处理库中的查找函数	308	12.1 引言	426
8.9 字符串处理库中的内存函数	314	12.2 自引用结构	427
8.10 字符串处理库中的其它函数	318	12.3 动态内存分配	427
专题：高级字符串操作练习	327	12.4 链表	429
第 9 章 格式化输入/输出	332	12.5 堆栈	437
9.1 引言	332		
9.2 流	332		

12.6 队列	442	15.12 单目作用域运算符	525
12.7 树	448	15.13 函数重载	526
第 13 章 预处理程序	476	15.14 连接说明	529
13.1 引言	476	15.15 函数模板	529
13.2 预处理指令 #include	476	第 16 章 类和数据抽象 (第一	
13.3 预处理指令 #define: 符号		部分)	539
常量	477	16.1 引言	539
13.4 预处理指令 #define: 宏	477	16.2 结构的定义	541
13.5 条件编译	479	16.3 访问结构的成员	541
13.6 预处理指令 #error 和 #program	480	16.4 用结构实现用户定义的类型	
13.7 运算符 # 和 #	480	Time	542
13.8 行号	481	16.5 用类实现抽象数据类型 Time	544
13.9 预定义的符号常量	481	16.6 类的作用域和访问类的成员	550
13.10 宏 assert	481	16.7 接口和实现的分离	550
第 14 章 高级话题	486	16.8 控制对成员的访问	553
14.1 引言	486	16.9 访问函数和工具函数	557
14.2 UNIX 和 DOS 系统中的输入/输出		16.10 初始化类的对象: 构造函数	558
重定向	486	16.11 在构造函数中使用默认参数	561
14.3 变长参数列表	487	16.12 析构函数的使用	561
14.4 使用命令行参数	489	16.13 调用析构函数和构造函数的	
14.5 对编译多个源文件程序的说明	490	时机	565
14.6 用 exit 和 atexit 终止程序的执行	492	16.14 数据成员和成员函数的使用	567
14.7 类型限定符 volatile	494	16.15 微妙的陷阱: 返回对私有数据成员的	
14.8 整数和浮点数常数的后缀	494	引用	572
14.9 再谈文件	494	16.16 拷贝逐个成员的默认赋值	
14.10 信号处理	497	方式	575
14.11 动态内存分配: 函数 calloc 和		16.17 软件的可重用性	576
realloc	499	第 17 章 类和数据抽象 (第二	
14.12 无条件转移: goto 语句	499	部分)	584
第 15 章 把 C++ 看作更好的 C	507	17.1 引言	584
15.1 引言	507	17.2 const 对象和 const 成员函数	584
15.2 C++ 的单行注释	508	17.3 复合: 把类作为其它类的	
15.3 C++ 的输入/输出流	509	成员	592
15.4 C++ 中的声明	510	17.4 友元函数和友元类	596
15.5 在 C++ 中建立新的数据类型	511	17.5 使用 this 指针	599
15.6 函数原型和类型检查	512	17.6 动态内存分配: 运算符 new 和	
15.7 内联函数	513	delete	604
15.8 引用参数	517	17.7 类的静态成员	605
15.9 const 限定符	521	17.8 数据抽象和信息隐藏	609
15.10 动态内存分配: new 和 delete		17.8.1 范例: 数组抽象数据类型	610
运算符	523	17.8.2 范例: 字符串抽象数据	
15.11 默认参数	524	类型	611

17.8.3 范例: 队列抽象数据类型	611	20.2 类型域和 switch 语句	704
17.9 容器类和递取类	612	20.3 虚函数	705
17.10 模板类	612	20.4 抽象基类和具体类	706
第 18 章 运算符重载	620	20.5 多态性	707
18.1 引言	620	20.6 实例研究: 利用多态性的工资单 系统	708
18.2 运算符重载的基本知识	621	20.7 新类和动态联编	718
18.3 运算符重载的限制	622	20.8 虚析构函数	718
18.4 用作类成员的运算符函数和用作友 元函数的运算符函数	623	20.9 实例研究: 继承接口和实现	719
18.5 重载流插入和流提取运算符	624	第 21 章 C++ 输入/输出流	730
18.6 重载单目运算符	627	21.1 引言	731
18.7 重载双目运算符	628	21.2 流	732
18.8 实例研究: 数组类 Array	628	21.2.1 iostream 类库的头文件	732
18.9 类型之间的转换	638	21.2.2 输入/输出流类和对象	732
18.10 实例研究: 字符串类 String	639	21.3 输出流	734
18.11 重载 ++ 和 --	649	21.3.1 流插入运算符	734
18.12 实例研究: 类 Date	650	21.3.2 连续使用流插入/提取运算符	736
第 19 章 继承	667	21.3.3 输出 char * 类型的变量	737
19.1 引言	667	21.3.4 用成员函数 put 输出字符和 put 函数的连续调用	737
19.2 基类和派生类	668	21.4 输入流	738
19.3 受保护的成员	670	21.4.1 流提取运算符	738
19.4 把基类指针强制转换为派生类 指针	670	21.4.2 成员函数 get 和 getline	740
19.5 使用成员函数	675	21.4.3 类 istream 中的其它成员函数 (peek、putback 和 ignore)	743
19.6 在派生类中重定义基类 成员	675	21.4.4 类型安全的 I/O	743
19.7 公有的、受保护的和私有的 基类	680	21.5 成员函数 read、gcount 和 write 的无格式 输入/输出	743
19.8 直接基类和间接基类	680	21.6 流操纵算子	745
19.9 在派生类中使用构造函数和析构 函数	680	21.6.1 整数流的基数: 流操纵算子 dec、 oct、hex 和 setbase	745
19.10 把派生类对象隐式转换为基类 对象	684	21.6.2 设置浮点数精度 (precision、 setprecision)	746
19.11 关于继承的软件工程	685	21.6.3 设置域宽 (setw、width)	747
19.12 复合与继承的比较	686	21.6.4 用户自定义的操纵算子	748
19.13 对象的“使用”关系和“知道” 关系	686	21.7 流格式状态	748
19.14 实例研究: 类 Point、Circle 和 Cylinder	687	21.7.1 格式状态标志	750
19.15 多重继承	694	21.7.2 尾数零和小数点 (ios:: showpoint)	751
第 20 章 虚函数和多态性	704	21.7.3 对齐 (ios:: left、ios:: right、ios:: internal)	751
20.1 引言	704	21.7.4 设置填充字符 (fill、setfill)	753
		21.7.5 整数流的基数 (ios::dec、ios::oct、	

ios::hex, ios::showbase)	754	21.9 用户自定义类型的 I/O	760
21.7.6 浮点数和科学记数法(ios::scientific, ios::fixed)	755	21.10 把输出流系到输入流上	762
21.7.7 大/小写控制(ios::uppercase)	756	附录 A C 语法	777
21.7.8 设置及清除格式标志 (flags, . setios-flags, resetiosflags)	757	附录 B 标准库	789
21.8 流错误状态	758	附录 C 运算符的优先级与结合性	818
		附录 D ASCII 字符集	819
		附录 E 数值系统	820