

Prentice-Hall

C/C++/Java 程序设计经典教程

(第三版)

[美] Harvey M. Deitel 著
Paul J. Deitel

贺民 译

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书详细介绍了 C 语言、C++ 以及 Java 语言。本书内容全面，代码实例丰富，除了对基本概念及编程方法进行了清晰而详尽的讲解外，还提供了作者本人在多年编程和教学工作中所收集的大量编程技巧和经验，常见的编程错误以及解决方法，以及在代码可移植性、程序性能提高、软件工程等方面的实用技巧。

全书共分为四个部分，第 1 部分是本书的重点，详细介绍了 C 语言编程各个方面的内容；第 2 部分介绍 C++ 编程，读者可以从中学掌握面向对象编程的要点；第 3 部分介绍 Java 编程，内容涉及到图形编程、使用 Java Swing 进行 GUI 设计、多媒体编程以及事件驱动的编程；第 4 部分是附录。

本书的作者具有丰富的计算机项目开发和教学经验，本书是作者多年来教学经验的积累，其内容安排与大学课程内容一致。

本书适用于初、中级的程序员。

C How to Program

Copyright © 2001 by The Prentice-Hall.

All rights reserved No part of the book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher. This edition is authorized for sale only in the People's Republic of China (excluding the special Administrative Region of Hong Kong and Macau).

本书中文简体字版由美国培生教育出版社集团授权清华大学出版社和北京科海培训中心合作出版。未经出版者书面允许不得以任何方式复制或抄袭本书内容。

版权所有，盗版必究。

本书封面贴有 Pearson Education 培生教育出版集团激光防伪标签，无标签者不得销售。

书 名：程序设计经典教程——C/C++/Java

作 者：Harvey M. Deitel, Paul J. Deitel

译 者：贺民 等

责任编辑：

出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）

<http://www.tup.tsinghua.edu.cn>

印 刷 者：北京门头沟胶印厂

发 行：新华书店总店北京科技发行所

开 本：787 × 1092 1/16 印张：86.75 字数：1 943 千字

版 次：2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

印 数：0000~5000

盘 号：ISBN 7-900643-42-7

定 价：128.00 元（ICD）

译者序

作为计算机专业开发人员，我们在使用程序设计语言方面走过了一段漫长的路，一段曲曲折折、布满经验和教训的路。在 80 年代，我们学习计算机课程时，通常都以 PASCAL 作为入门语言，那时，C 语言还只是作为一门选修课。但毕业走入工作岗位后，却发现实际上 C 语言才真正大有用武之地，在我们开发的许多项目中，都使用 C 语言作为主要的开发语言。

此后，我们一直关注有关 C 语言的书，也买了和看了许多，然而，要找一本合适的书却并没有那么容易。有些 C 语言的编程书尽管很有名气，许多大学都在用它作为教科书，但由于出书时的环境所限，其中有些内容已经相当陈旧甚至早就过时了。

我们希望有一本很实用的书，它具有如下特点：

- 基本概念叙述清晰，并通过实例让读者掌握语言。
- 叙述的内容是最新的。
- 给出程序员在编程时经常遇到的错误，并告诉你如何避免这些问题。
- 不仅有清晰的概念叙述，还有程序员丰富的经验之谈。
- 有大量的编程技巧，在程序的分析中融合了软件工程的原理。
- 大量的源代码分析，让初学者可以轻松掌握编程技巧。
- 有大量的练习和解答。
- 作者具有丰富的软件开发和教学经验，是权威的教师或软件开发人员。

本书就是这样一本非常优秀的参考书，它能够满足你在 C 语言编程中几乎所有需要。它全面、综合、完整、清晰，实用性强。另外，本书还介绍了 C++ 和 Java 编程，这些也是现在的计算机编程中相当流行的编程语言。

为了使层次清晰，我们将原书按内容分为如下 4 部分（原书没有按部分划分章节）：

- C 语言编程
- C++ 编程
- Java 编程
- 附录

本书翻译人员均为多年从事程序开发工作的程序员。翻译过程中，我们更正了原书中的一些疏漏之处。希望我们能够完整地体现原书的精髓，将一本非常有特色的好书完好地呈现在读者面前。

译者

2001 年 8 月

前 言

欢迎学习 ANSI/ISO Standard C，欢迎学习 C++和 Java。本书是由一位老人和一位年轻人共同完成的。这位老人（H. M. Deitel，1967年毕业于美国麻省理工学院）从事编程和教授编程的工作已经有 39 年了；这位年轻人（P. J. Deital，1991年毕业于美国麻省理工学院）从事编程工作也有 18 年了，他发现了教学和写作中的一些漏洞。老人根据经验来编程和教学，年轻人依靠充沛的体力来做这些工作。老人追求程序的清晰，年轻人关注程序的性能。老人欣赏优雅和美丽，年轻人渴望的是结果。我们两个人携手共同编写了这本书，我们希望您将会发现这本书内容丰富、有趣，读起来轻松愉快。

编写本书的原因

在多数教育环境中，C 语言被教授给那些知道如何编程的人。很多教育者认为，由于 C 语言的复杂性以及很多其他困难，所以，在第一次学习编程时，不应当学习 C 语言，而这正好是本书的目标。我们为什么要编写这本书呢？

Harvey M. Deitel 博士在大学中讲授了 20 多年的编程入门课程，他所强调的重点是开发编写清晰而且结构很好的程序。在这些课程中，很多内容都是结构化编程的基本原理，并强调了控制结构和功能的有效使用。我们按照与 Harvey M. Deitel 在大学课程中完全相同的方式来介绍这些材料。如果学生们知道自己所学的语言在离开大学环境后就可以立即使用的话，就会有很高的学习热情。

我们的目的非常清楚：为那些只有很少甚至没有编程经验的学生编写一本 C 程序设计课本，它可以作为计算机编程方面的大学入门级课程使用，同时，本书对传统的 C 语言课程所要求的理论和实践进行深入而准确的讲解。为了实现这些目标，我们编写了一本比其他 C 语言课本更厚的书，这是因为，我们将在本书中耐心地讲授结构化编程的原理。全世界有成千上万的学生通过本书的早期版本来学习 C 语言。

本书包含了从很多领域中收集的丰富的实例、练习以及项目，目的是让学生能够有机会来解决现实世界中的有趣问题。本书还集中讲述了良好的软件工程的原理，并通过结构化编程方法的使用来强调程序的清晰性。我们是通过实例来介绍知识的。

本书第三版的说明

本书内容已经经过了许多业界和学术专家的仔细检查，其中包括 ANSIC 委员会的首脑。我们已经对第二版中的材料进行了提炼，尤其是关于指针的那一章。

在本书第二版中，我们用了 7 章的篇幅来介绍 C++ 和面向对象编程。在第二版出版以后，很多大学决定，把介绍 C++ 和面向对象编程的内容结合到 C 语言课程中。所以，在这个版本中，我们把 C++ 的内容扩充为 9 章，其中包含了一个学期课程所需要的足够的内容、练习和实验。我们还添加了 7 章使用 Java 进行面向对象编程的内容，其中包括了图形编程、使用 Java Swing 组件的图形用户界面（GUI）以及多媒体编程的内容。

1999 年，国际标准化组织批准了 C 语言的一个新版本，即 C99。但在编写这本书的时候，还没有任何 C99 编译程序可用。因此，我们无法使代码实例与 C99 兼容。在可以使用 C99 编译程序时，我们将测试本书中的每个程序，并在我们的 Web 站点（www.deitel.com）上列出所有的不同之处。我们还将在该 Web 站点张贴解释 C99 新特性的代码实例。附录 B 中包含了 Internet 和 WWW 上 C99 资源的完整清单。如果想得到关于 C99 的更多信息以及购买标准文档（ISO/IEC 9899:1999）的副本，请访问美国国家标准化组织（ANSI）的 Web 站点（www.ansi.org）。

本书第三版的辅助工具

我们努力工作，编写了这本书及其辅助资料，我们希望教师和学生都会发现这些材料非常有价值。我们非常感谢 Borland 公司为本书提供的产品，这些产品都包含在本书配套光盘中，它们使读者可以编译和运行用 C、C++ 和 Java 语言编写的程序。另外，还有下面这些资源可供利用：

- 本书配套光盘中包含本书中的 270 个程序实例，这些实例也可以从 www.deitel.com 站点下载。这有助于教师更快地准备讲义，也有助于学生掌握 C 语言。当把这些源代码从 ZIP 文件中解压缩时，必须使用诸如 WinZip（www.winzip.com）或 PKZIP（www.pkzip.com）这些能够理解目录的 ZIP 文件阅读程序。文件会被解压缩到一个单独的目录中（如 `chtp3e_examples`）。学生应该编译和执行他们在课本中学到的每个程序。
- 本书配套光盘中还提供了完整的 Borland C++ 编译程序 5.5 版本。这个软件使学生能够从命令行中编辑、编译和调试 C 以及 C++ 程序。光盘中包含了 Borland C++ Builder 5 的 60 天试用版，该产品提供了完整的集成 C/C++ 开发环境。
- 本书配套光盘提供了 JBuilder 3.5 的试用版。该软件使学生能够在集成开发环境中编辑、编译和调试 Java 程序。
- Instructor's Manual（教师手册）光盘中有本书多数练习的解答。程序按照章节和练习编号分在各个目录中（注意，请不要给我们写信索取教师光盘。这个光盘的发布被严

格限制在使用本书进行教学的大学教授的范围。教师只能通过他们的 Prentice Hall 代表来获取该解答手册)。

- 配套 Web 站点 (www.prenhall.com/deitel) 中也提供了教师和学生资源。教师资源包括课本附录 (如附录 A) 和用于课程计划的大纲管理器。学生资源包括章节目标、带有解答的判断对错类型的复习题、参考材料和留言板。
- 可定制的 Powerpoint Instructor Lecture Notes (教师讲义笔记), 其中包括源代码, 以及对每个程序及重要图解的关键讨论。教师和学生可以在我们的 Web 站点 www.deitel.com 上免费获得这些讲义笔记。尽管教师可以修改这些笔记, 并把它们用于课堂讲解中, 但必须知道, 这些笔记是 Prentice Hall 公司版权所有的, 如果没有得到该公司明确的书面许可, 任何人都不能使用这些笔记。

本书说明

本书包含了从很多领域内挑选出来的丰富的实例、练习和项目, 其目的是为学生提供一个解决现实世界问题的机会。本书集中讲述了良好的软件工程原理, 并强调了程序的清晰性。我们是通过实例进行讲解的。

本书的作者曾经花费大量的时间来教授和写作编程语言。本书的重点是教学。例如, 实际上, 无论是 C、C++ 还是 Java 的新概念, 都是通过完整的可运行的程序进行讲解的, 并在该程序之后还附加了一个窗口, 来显示该程序的输入和输出。读这些程序的过程就好像是在计算机上输入并运行它们一样。我们把这叫作“实时代码方法”。

本书的其他特点还有: 每章开头列出了本章的目标; 常见的编程错误、良好的编程习惯、性能提示、可移植性提示、软件工程经验以及测试和调试提示, 这些内容都在每章的结尾部分列举和汇总; 每章都全面总结所学的内容, 并列出了所涉及的术语; 每章都有自测练习; 还有所有 C 语言书籍中最丰富的练习集。

这些练习的内容包括从简单地回忆问题, 到冗长的编程问题, 甚至涉及到一些较大的项目。需要基本学期项目的教师将发现, 这些练习中所列出的很多问题都适合自己的需要, 本书后面的章节尤其如此。我们在练习方面做出了很大的努力, 其目的是为了使学生能够从本课程中真正掌握知识。

在编写本书时, 我们使用了多种 C 编译程序。通常, 本书中的程序可以在所有 ANSI/ISO C 编译程序上运行, 其中包括本书所带的 Borland C++ 编译程序。

本书有关 C 语言的内容 (从第 2 章到第 14 章) 都遵循 1990 年发布的 ANSI C 标准。ANSI C 的很多功能不能与早期的 ANSI C 版本一起工作, 可能也无法与 ISO C99 编译程序一起工作 (当可以使用 C99 编译程序的时候)。请查看自己特定系统的参考手册, 以获取有关该语言的更多细节, 或者从美国国家标准化组织索取 ANSI/ISO 9899:1990 “American National Standard for Information Systems—Programming Language C (信息系统的美国国家标准—编程语言 C)”

的副本，该组织的地址是 11 West 42nd Street, New York, New York 10036。

C++的内容以 C++编程语言为基础，它们分别由委任标准委员会 X3 (Accredited Standards Committee X3) 的信息技术分会 (Information Technology) 及其技术委员会 X3J16 (Technical Committee X3J16) 的编程语言 C 分会开发。C 和 C++语言得到了国际标准化组织 (ISO) 的批准。如果想得到更详细的信息，请与下面的地址联系：

```
X3 Secretariat  
1250 Eye Street NW  
Washington DC 20005
```

严谨的程序员应该仔细阅读这些文档，并经常参考它们。这些文档并不是教程。它们使用编译程序实现者和“繁忙”的开发人员所要求的特别精确度来定义各自的语言。

Java 章节中的内容以 Sun Microsystem 公司最新的 Java 版本为基础，即 Java 2 Platform。Sun 公司提供了名为 Java 2 Software Development Kit (J2SDK, Java 2 软件开发工具包) 的 Java 2 Platform 实现方法，其 1.3 版包括需要用来编写 Java 软件的最少的工具集。如果想深入了解最新 Java 实现方法的细节，请访问 java.sun.com 站点。

我们仔细按照这些文档和资料审查了本书的内容。我们这本书适用于初中级用户。我们没有试图涵盖这些综合性的文档中所讨论的每一个功能。

目标

每章开头都会提出本章的目标。这可以告诉学生应该掌握什么，并给学生们一个机会，使他们在读完这一章之后能够判断出自己是否达到了这些目标。这可以增强学生的自信，帮助他们积极巩固所学的内容。

270个实例程序，12,178行代码，包括程序输出

我们使用可运行的完整程序来介绍 C、C++和 Java 语言的功能，每个程序后面都提供了一个窗口，其中包含了该程序运行时产生的输出结果，我们把这叫作“实时代码方法”。这使学生可以确认程序的运行是否正确。把输出结果放在产生输出结果的程序语句的后面，是学习和巩固概念的好方法。我们的程序对 C、C++和 Java 的不同功能进行练习。仔细阅读本书的过程就好像是在计算机上输入和运行这些程序一样。

473个图表/图

本书提供了丰富的图表和线条图。在第 3 章和第 4 章中，我们讨论控制结构时，利用仔细绘制的流程图进行讲解(请注意，我们没有把流程图的使用作为一种程序开发的工具来介绍，但我们的确使用了面向流程图的简短介绍，来指定 C 语言控制结构的精确操作)。第 12 章使用线条图来说明连接列表、队列、堆栈以及二叉树的创建和维护。本书的其余部分也提供了详尽的图解说明。

784个编程提示

本书包含了6个设计要素，来帮助学生们把注意力集中于程序开发、测试和调试、性能和可移植性的重要方面。我们以“良好的编程习惯”、“常见的编程错误”、“性能提示”、“可移植性提示”、“软件工程经验”和“测试和调试提示”的形式来强调了上百条这样的提示。这些提示信息是我们从近60年来编程和教学经验中所收集到的最好的经验。我们的一个学生（数学专业）最近告诉我们，她感觉这种方法与数学课本中对公理、定理和推论进行强调的做法有些类似，它为构建优秀软件提供了基础。



163 个良好的编程习惯

良好的编程习惯在文中进行了突出显示。它们能够使学生们更关注那些有助于编写出更好程序的技术。当我们向非程序员讲授入门课程时，我们在每堂课上不停地强调“清晰性”，我们告诉学生，我们将在“良好的编程习惯”中强调这些用于编写出更清楚、更易理解和更易维护的技术。



262 个常见的编程错误

学习一门语言的学生（尤其是初次学习程序设计的学生）通常会出现各种错误。集中关注这些常见的编程错误，可以帮助学生避免犯同样的错误。它还有助于减少学生向教师提问的次数。



76 个性能提示

根据我们的经验，对于第一门程序设计课程而言，教学生编写出清晰的、可理解的程序是最重要的目标。但学生们希望写出运行最快、使用内存最少、按键次数最少或者在其他方面非常突出的程序。实际上，学生们关心的是程序的性能。他们想知道自己能够进行哪些操作让程序性能更好。所以，我们在本书中包含了性能提示，以特别强调那些能够提高程序性能的方法。



41 个可移植性提示

软件开发是复杂而昂贵的活动。开发软件的企业必须经常生产出适于各种计算机和操作系统的软件版本。所以，今天人们非常重视软件的可移植性，也就是说，生产出只需进行很少修改甚至无需修改就能够在各种不同计算机系统上运行的软件。很多人把C/C++誉为用于开发可移植程序的合适语言，尤其是因为ANSI/ISO C和C++是这两种语言的全球标准。有些人认为，如果他们执行用C++编写的应用程序，那么该应用程序将自动成为可移植的。实际情况却不是这样。实现可移植性需要经过认真仔细的设计。在这个过程中，有很多陷阱。我们在本书中包括了很多可移植性方面的提示，以帮助学生编写可移植的代码。Java语言的设计目的是为了实现在最大的可移植性，但为了移植它们，Java程序还是需要进行修改。



187 个软件工程经验

软件工程经验强调了那些影响软件系统（尤其是大型软件系统）体系结构和构建的技术、结构问题和设计问题等。在高级课程中和学生们开始针对大型复杂的现实系统而工作时，从本书中学到的很多内容都会很有用。C、C++和 Java 是实现良好的软件工程特别有效的语言。

23 个测试和调试提示



这个名称也许不太恰当。当我们决定把测试和调试提示放到这个新版本中时，我们认为这些信息将会对测试程序以发现错误会有所提示，并对删除这些错误提出建议。实际上，这些提示中的大多数是对 C、C++和 Java 中那些防止错误进入程序中的功能和特性的观察。

32 个外观和感觉经验



在本书的 Java 部分，我们提供了这类经验来强调图形用户界面方面的约定。这些经验能够帮助学生们设计出符合行业标准的图形用户界面。

728个自测练习题和解答（包括独立的部分）

本书中还包含了大量自测练习和自测练习的解答供读者自学。这给学生们提供了使用这些资料建立信心和尝试这些常规练习的机会。

994个练习题（包括独立的部分，总共1722个练习题）

每章最后都提供了大量的练习题，这些练习题包括对重要术语和概念的回忆：编写单个程序语句；编写一小部分函数和 C++/Java 类；编写完整的函数、C++/Java 类程序；以及编写主要的学期项目。大量的练习使教师能够调整自己的课程，以满足听众独特的需要和每学期不同的课程安排。教师可以使用这些练习来布置家庭作业、安排小测验或重要的考试。

550页带有练习解答的教师手册

在教师光盘和通过 Prentice Hall 的代表而得到的教师专用磁盘中，包含有这些练习的解答（注意，请不要写信给我们来索取教师光盘。这个光盘的发布对象被严格限制为使用本书进行教学的大学教授。教师只能通过他们的 Prentice Hall 的代表来得到这个解答手册）。

本书主要内容

本书分为 4 部分。第一部分（从第 1 章到第 14 章）介绍了 C 语言的全部知识，其中包括对结构化程序设计的正式介绍。第二部分（从第 15 章到第 23 章）是所有 C 语言课本中独一无二的，这部分对 C++和面向对象的程序设计进行了实质性的讲解，这对更高级别的本科大学课程而言绰绰有余。第三部分（从第 24 章到第 30 章）在所有 C 语言课本中也是独一无二的，它对 Java 进行了完整的介绍，其中包括图形编程、使用 Java Swing 进行 GUI 设计、多媒体编程和事件驱动的编程。第四部分（从附录 A 到附录 E）介绍了与本课本相关的大量参考

材料。

第 1 章“计算机、Internet 和万维网入门”讨论了计算机是什么，计算机如何工作以及计算机是如何编程的。它介绍了结构化编程的概念，并解释了为什么这种技术引起了程序编写方法的革命。本章从机器语言开始，逐步介绍了汇编语言和高级语言，提供了编程语言开发的简短历史。本章还讨论了 C、C++ 和 Java 编程语言的起源。本章中包括了对典型 C 语言编程环境的介绍。我们还讨论了，随着 WWW 和 Java 编程语言的出现，人们对 Internet 的兴趣大大提高。

第 2 章“C 程序设计入门”简要介绍 C 程序的编写，并详细讲解了 C 语言中的决策生成和算术运算。学完本章之后，将会了解如何编写出简单而完整的 C 程序。

第 3 章“C 语言结构化程序开发”可能是本书中最重要的一章，尤其是对于计算机科学专业那些严谨的学生而言。它介绍了用于解决问题的算法（过程）的概念，还解释了结构化程序设计的开发易于理解、易于调试、易于维护以及第一次就能够正常运行的程序中的重要性。本章介绍了结构化程序设计的基础控制结构，也就是顺序结构、选择（if 和 if/else）结构和循环（while）结构。它还解释了自上而下、逐步求精的技术，这对编写出正确的结构化程序而言是至关重要的。本章介绍了流行的程序设计工具：结构化伪代码。第 3 章所用的方法对于任何编程语言的结构化程序设计都适用，而不仅仅是 C 语言。本章帮助学生养成良好的程序设计习惯，为他们处理课本余下部分中的主要程序设计任务奠定基础。

第 4 章“C 程序控制”详细介绍了结构化程序设计的概念，并介绍了其他控制结构。本章详细地讲解了循环结构，并比较了计数器控制的循环和标记控制的循环。for 结构为实现计数器控制的循环的一种便利方法。本章还介绍了 switch 选择结构和 do/while 循环结构。这一章的最后讨论了逻辑运算符。

第 5 章“C 函数”讨论了程序模块的设计和构建。C 语言与函数相关的功能包括标准库函数、程序员定义的函数、递归和按值调用的功能。第 5 章中介绍的技术对于编写和评价正确的结构化程序而言是非常重要的，尤其是对于那些系统程序员和应用程序设计者在现实应用中开发的大型程序和软件。通过把复杂的问题划分为多个较小的相互作用的组件，这种“分而治之”的战略被认为是解决复杂问题的一种有效途径。学生们喜欢本章中对随机数和模拟程序的讲解，他们还喜欢本章对掷骰子游戏的讨论，在这个游戏中非常好地使用了控制结构。本章还详细介绍了递归，并利用一个表格总结了本书其余部分中的数十个递归实例和练习。有些书把递归放在全书内容的后面，但我们感觉这个主题通常会在整本书中涉及到。第 5 章后面的大量练习中包括了几个经典的递归问题（如汉诺塔问题）。

第 6 章“C 数组”讨论了相同类型的相关数据项的数据结构。本章介绍了大量单下标数组和双下标数组的实例。人们普遍认为，在正确的结构化程序的开发过程中，正确地结构化数据与有效地使用控制结构一样重要。本书中的实例考察了各种常见的数据控制、打印直方图、排序数据、把数组传递到函数中，以及对调查数据分析领域的介绍（使用采样数据）。本章的一个特点就是详细介绍了基本的排序和搜索技术，并把对分查找方法作为一种在线性搜索基础上

快速提高效率的搜索方法来介绍。本章结尾部分的练习包括了很多有趣的具有挑战性的问题，如改进的排序技术、机票预订系统的设计、龟图概念的介绍（在 LOGO 语言中非常著名），以及骑士旅行和 8 皇后问题，该问题介绍了在人工智能领域内广泛应用的启发式编程的概念。

第 7 章“C 指针”介绍了 C 语言中功能最强大的和最难掌握的功能，也就是指针。本章详细介绍了指针运算符、按引用调用、指针表达式、指针算法、指针和数组的关系、指针的数据和用于函数中的指针等知识。本章练习中包括了一个古典的龟兔赛跑的模拟，还包括了洗牌和发牌算法以及递归的走迷宫问题。这里还包括了一个名为“构建自己的计算机”这一特殊小节。本节解释了机器语言编程，接着介绍了一个包含计算机模拟程序的设计和实现的项目，在这个模拟程序中，读者可以编写和运行机器语言程序。本书中的这个特色是独一无二的，对于需要理解计算机是如何实际工作的读者来说特别有用。我们的学生都非常喜欢这个项目，并且经常对其实现了增强功能，很多功能都是从这些练习中提出的。在第 12 章中，有另外一个小节会指导读者去建立一个编译程序，然后这个编译程序生成的机器语言就会在第 7 章所编写的机器语言模拟程序上执行。

第 8 章“C 字符和字符串”介绍了处理非数字数据的基础知识。本章完整地介绍了 C 语言库中可用的字符和字符串处理函数。这里所讨论的技术广泛应用于构建字处理程序、页面排版和排版软件以及字处理应用程序中。本章包括一个有趣的练习集，这些练习研究了字处理应用程序。学生们非常喜欢这些练习，这些练习包括写作五行打油诗、写作任意诗歌、把英语转换为拉丁语、生成与给定电话号码等效的 7 个字母的单词、文本对齐、检查保护、检验单词数量、生成摩尔斯电码、公制转换以及催讨信函等。对学生具有挑战意义的最后一个练习是使用计算机字典去创建一个纵横字谜的生成器。

第 9 章“C 格式化输入/输出”介绍了 `printf` 和 `scanf` 所有强大的格式化功能。我们讨论了 `printf` 的输出格式化功能，如把浮点值舍入到给定的小数位、对齐数字列、右对齐和左对齐、文字信息的插入、强制使用加号、打印前导零、使用指数表示、使用八进制数和十六进制数，以及控制字段的宽度和精确度。对于光标移动、打印特殊字符以及产生可听见的警报声音，我们讨论了 `printf` 转义序列的所有知识。我们介绍了 `scanf` 的输入格式化功能，包括输入特殊类型的数据，在输入流中跳过特殊的字符。我们讨论了 `scanf` 所有用于读取十进制、八进制、十六进制、浮点、字符和字符串值的转换说明符。我们还讨论了扫描输入以匹配（或不匹配）扫描集中的字符。本章中的练习实际上对 C 语言所有的格式化输入/输出功能进行了测试。

第 10 章“C 结构、联合、位操作和枚举”介绍了各种重要的功能。结构就像 Pascal 和其他编程语言中的记录一样，它们能够将各种类型的数据项组合在一起。第 11 章中使用结构来形成那些由信息记录构成的文件。结构与指针以及第 12 章中的动态内存分配结合在一起使用，可以形成诸如链接表、队列、堆栈和树这样的数据结构。联合使内存中的一个区域能够在不同的时间用于不同的数据类型，这样的共享方式能够减少程序的内存需求或辅助存储器的需求。枚举提供了定义有用的符号常量的方便途径，这有助于使程序成为自说明的。C 语言强大的位操作能力使程序员能够编写出利用低级硬件功能的程序。这有助于程序处理位字符串、设置单

独位的开或关状态，以及更加紧凑地存储信息。这样的功能通常只能在低级汇编语言中找到，它们得到了那些编写诸如操作系统和连网软件等系统软件的程序员的重视。本章的一个特色就是，它的洗牌和发牌模拟程序经过了修订，具有很高的性能。这对于那些强调算法质量的教师而言，是一个非常好的机会。

第 11 章“C 文件处理”讨论了以顺序访问和随机访问方式处理文本文件的技术。本章开始按照从位到字节、然后到字段和记录、最后到文件的顺序对数据的层次结构进行了介绍。接着介绍了 C 语言文件和流的简单概念。我们使用程序讨论了顺序访问文件，这些程序显示了如何打开和关闭文件、如何在文件中按顺序存储数据，以及如何从文件中按顺序读取数据。我们也使用程序讨论随机访问文件，这些程序显示了如何从随机访问的文件中按顺序读数据。第 4 个随机访问的程序把很多按顺序和随机方式访问文件的技术综合到了一个完整的事务处理程序中。研究班上的学生告诉我们，在学完文件处理方面的材料之后，他们能够编写出一些在自己企业中可立即使用的真实的文件处理程序。

第 12 章“C 数据结构”讨论了用于创建和控制动态数据结构的技术。本章开头讨论了自引用结构和动态内存分配，接着讨论了如何创建和维护各种动态数据结构，其中包括链接表、队列（或等待队列）、堆栈和树。对于每种类型的数据结构，我们介绍了可以运行的完整程序，并显示了示范的输出结果。本章能够帮助学生掌握指针。本章还包括了大量使用间接和双间接（一个特别难以理解的概念）的实例。使用指针的一个问题就是，学生们在想像数据结构以及它们的节点如何连接在一起的时候存在着困难。所以我们在本章中包括了一些图解，来显示这些链接以及创建这些链接的顺序。二叉树的实例是学习指针和动态数据结构的最好工具。这个实例创建了一个二叉树，加强了重复消除，引入了递归的前序、中序和后序树遍历。当学生们学习和实现这个实例时，他们会真正获得一种成就感。他们尤其喜欢看到中序遍历按照排序后的顺序打印出节点值。本章还包括了一个练习集。这些练习的突出部分就是“构建自己的编译程序”这一特殊章节。这些练习还带领着学生了解了一个前缀后缀转换程序和一个后缀表达式赋值程序的开发过程。然后，我们调整了这个后缀赋值算法以生成机器语言代码。编译程序把该代码放在一个文件中（使用第 11 章介绍的技术）。然后，学生们就可以在它们从第 7 章练习中所创建的软件模拟程序上运行编译程序所生成的机器语言。

第 13 章“C 预处理器”详细讨论了预处理器的指令。本章中包括了 `#include` 伪指令和 `#define` 伪指令的详细信息，其中 `#include` 伪指令会在文件编译之前把指定文件的一个拷贝包含在该伪指令的位置，`#define` 伪指令用来创建符号常量和宏。本章还解释了条件编译，条件编译使程序员能够控制预处理伪指令的执行和程序代码的编译。本章讨论了 `#运算符` 和 `##运算符`，`#运算符` 可以把它的操作数转换为字符串，`##运算符` 可以连接两个标记。我们还介绍了预定义的符号常量 `_LINE_`、`_FILE_`、`_DATE_` 和 `_TIME_`。本章的最后讨论了 `assert.h` 头文件的宏 `assert`。在程序测试、调试、验证和确认中，宏 `assert` 是非常有价值的。

第 14 章“C 语言高级技术”介绍了其他一些主题，其中包括几个在入门课程中通常没有介绍的高级主题。我们说明了：如何把程序的输入重定向为来自一个文件，把程序的输出重定

向到一个文件中，把程序的输出重定向为另一个程序的输入（这叫作管道），把一个程序的输入添加到现有的文件中，开发出使用变长参数列表的函数，把命令行参数传递到 `main` 函数中并在程序中使用它们，对组件分布在多个文件中的程序进行编译，使用 `atexit` 注册函数使其能够在程序终止时执行，使用函数 `exit` 终止程序的执行，使用 `const` 和 `volatile` 类型限定符，指定使用整数或浮点后缀的数字式常量的类型，使用信号处理库来捕获意外事件，使用 `calloc` 和 `realloc` 来创建和使用动态数组，以及把 `union` 作为一种节省空间的技术。

第 15 章“C 的增强版本 C++”介绍了 C++ 的非面向对象的功能。这些功能改进了编写过程程序的进程。本章讨论了单行注释、流输入/输出、声明、创建新的数据类型、函数原型和输入检查、`inline` 函数（作为宏的替代）、引用参数、`const` 限定符、动态内存分配、默认参数、一元作用域解析运算符、函数重载、链接规范和函数模板。

第 16 章“C++ 类和数据抽象”开始讨论基于对象的编程。本章非常适于学习数据抽象，即通过一种专门用于实现抽象数据类型（ADT）的语言（C++）。近年来，数据抽象已经成为入门级计算机课程的主要主题。从第 16 章到第 18 章的内容详细讲解了数据抽象。第 16 章讨论了把 ADT 实现为 C++ 样式的类以及这种方法为什么比结构好、访问类成员、从实现中分离接口、使用访问函数和实用函数、使用构造函数初始化对象、使用析构函数破坏对象，以及通过默认的按成员复制和软件可重用性进行赋值等。本章中的一个练习要求读者为复杂的数字开发出一个类。

第 17 章“C++ 类：第二部分”继续研究类和数据抽象。本章讨论了声明和使用常量对象、常量成员函数、构成（构建把其他类的对象作为自己成员的类的过程）、对类的 `private` 和 `protected` 成员具有特殊访问权限的 `friend` 函数和 `friend` 类、使对象知道自己地址的 `this` 指针、动态内存分配、用于维护和处理整个类范围内的数据的 `static` 类成员、抽象数据类型（数组、字符串和队列）的实例、容器类和迭代。本章中的练习要求学生开发出一个存款账户类和一个用于保存多组整数的类。我们讨论了使用 `new` 和 `delete` 进行动态内存分配。当 `new` 失败时，在以前的标准 C++ 中，它会返回一个 0 指针。我们在从第 17 章到第 22 章的内容中使用了这种以前的标准样式。我们推迟到第 23 章讨论新样式的 `new` 故障，在这种情况下，`new` 会发出一个异常。我们通过基于视频游戏的实例讨论 `static` 类成员。我们在整本书中和专业研讨班中始终强调，对类的客户程序隐藏实现的细节是非常重要的。

第 18 章“C++ 运算符重载”是 C++ 课程中最受欢迎的主题之一。学生们很喜欢这部分材料。他们发现本章内容与第 16 章和第 17 章中对抽象数据类型讨论非常相符。运算符重载使程序员能够告诉编译程序，如何把现有的运算符与新的类对象一起使用。C++ 已经知道如何把这些运算符与诸如整数、浮点数和字符这样的内置类型的对象一起使用。但是，假设我们创建了一个新的字符串类，那么，当用于字符串对象之间时，加号究竟表示什么？很多程序员把加号用在字符串中，以此来表示连接。本章讨论了运算符重载、运算符重载中的限制、类成员函数重载对非成员函数重载、重载一元运算符和重载二元运算符，以及这两种类型之间的转换等方面的基础知识。本章的一个特点就是实际案例来学习数组类、巨大的整数类和复杂的数字

类（后面两种在练习中有足够的源代码）。这个材料与您在多数编程语言和课程中遇到的不同。运算符重载是复杂的主题，但也是非常吸引人的主题。聪明地使用运算符重载可以使类变得格外“优雅”。与第 16 章、第 17 章和第 18 章中的技术一起使用，我们就能够精巧地创建一个 `Date` 类（如果在过去 20 年中我们一直在使用它的话），这个类能够非常容易地解决“2000 年问题”中的主要问题。本章中的一个练习鼓励读者向 `complex` 类中添加运算符重载，使其能够通过运算符对这个类的对象进行方便的处理，像数学中那样，而不是通过学生们在第 17 章练习中所用的函数调用方法。

第 19 章“C++继承”讲解了面向对象编程语言的最基础的功能之一。继承是软件可用的一种形式，在这种形式中，通过吸收现有类的功能并添加合适的新功能，就能够快捷容易地开发出新的类。本章讨论了基类和派生类的概念、`protected` 成员、`public` 继承、`protected` 继承、`private` 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，以及带有继承的软件工程。本章比较了继承（“是一种”关系）和组合（“具有一种”关系），并介绍了“使用一种”关系和“了解一种”关系。本章的一个特点就是它用几个实际案例教学。尤其是有一个较长的案例能够实现点、圆和圆柱类的继承。这些练习要求学生去对比通过继承来创建新类和通过组合来创建新类；扩展本章中所讨论的各种继承层次；为四边形、梯形、平行四边形、矩形和正方形编写一个继承的层次结构；使用二维形状和三维形状创建更加通用的形状继承。

第 20 章“C++虚拟函数和多态性”讲解了面向对象程序设计的另外一个继承功能，也就是多态性行为。当很多类通过继承关联到普通的基类中时，每个派生类对象都可以作为基类对象。这使程序能够按照独立于特定类型的派生类对象的通用方式进行编写。新类型的对象能够由相同的程序进行处理，因此使系统更加具有可扩展性。多态性使程序能够消除复杂的 `switch` 逻辑而支持更加简单的“直线型”逻辑。例如，视频游戏的屏幕管理程序能够向待绘制的对象链接表中的每个对象简单地发送绘图信息。每个对象都知道如何绘制出自己的图形。只要新的对象也知道如何绘制自己的图形，那么，无需修改该程序，就可以把这个新对象添加到程序中。这种编程类型通常用于实现今天大受欢迎的图形用户界面（GUI）。本章讨论了通过使用虚拟函数来实现多态性行为的机制。它分为抽象类（在抽象类中，对象不能初始化）和具体类（在具体类中，对象能够初始化）两类。对于为整个层次结构中的类提供一个可继承的接口，抽象类是非常有用的。本章的一个特点就是，它对第 19 章中所讨论的点、圆、圆柱形层次结构进行了多态性的案例学习。本章中的练习要求学生去讨论大量概念性的问题和方法、向形状层次结构中添加抽象类、开发基础的图形包，使用虚拟函数和多态性程序设计来完成所有这些项目。我们的专业读者坚持要求我们应该提供更加深入的解释，以精确地表明如何在 C++ 语言中实现多态性，因此，我们明确介绍了使用这种功能编程时设计者应该花费的执行时间以及内存需要。我们在“多态性、虚拟函数和动态绑定揭秘”这一小节中提供了一个图解方式，图解显示了 C++ 编译程序自动建立用以支持多态性程序设计样式的 `vtables`（虚表）。我们在讨论点、圆、圆柱形状层次结构的类中绘制了这些表格。我们的读者表示，对于他们将处理的每个新对象而言，这种做法的确给他们提供了一些信息，用以决定多态性是否是一种合适的程序设计样式。

我们在 20.9 节中介绍这方面的内容，并在图 20.2 中显示了 `vtable` 的图解。请认真学习这个内容。它会使您对程序中带有继承和多态性时计算机中实际发生的操作有更加深入的理解。

第 21 章“C++流输入/输出”全面而综合地讲解 C++面向对象的输入/输出。本章讨论了 C++的各种 I/O 功能，其中包括使用流插入运算符输出、使用流提取运算符输入、类型安全的 I/O（C 语言的很好改进）、格式化的 I/O、非格式化的 I/O（对于性能而言）、用于控制流基数的流处理程序（十进制、八进制或十六进制）、浮点数、控制字段宽度、用户定义的处理程序、流的格式状态、流的错误状态、用户定义的对象类型的 I/O，以及把输出流和输入流连接在一起（确保在期待用户输入响应之前出现提示）。本章的大量练习要求学生编写出对本章所讨论的大多数 I/O 功能进行测试的各种程序。

第 22 章“C++模板”讨论了 C++新增加的功能。在第 15 章中介绍了函数模板。类模板使程序员能够抓住抽象数据类型（如堆栈、数组或队列）的精髓，然后使用最少的额外代码为特定类型（诸如 `int` 队列、`float` 队列、字符串队列等）创建出该 ADT 的程序方案。由于这个原因，模板类通常叫作参数化类型。本章还讨论了标准参数和非标准参数的使用，并且考虑了模板与其他 C++概念（如继承、`friends` 和 `static` 成员）之间的互相作用。本章练习要求学生编写出各种函数模板和类模板，并在完整的程序中使用这些模板。

第 23 章“C++异常处理”讨论了 C++语言较新的增强功能之一。异常事件使程序员能够编写出更加稳定、具有更高容错能力以及更加适合于商业密集型和任务密集型环境中的程序。本章讨论了什么时候适于使用异常事件处理程序；介绍了使用 `try` 块、`throw` 语句和 `catch` 块的基本异常事件处理；指出了如何以及何时排除异常事件；解释了如何编写异常事件规范以及处理没有预料到的异常事件；还讨论了异常事件和构造函数、析构函数以及继承之间的重要联系。我们讨论了如何发出异常事件，并图解说明了当内存耗尽时，`new` 可能失效的两种方式。在 C++草案标准之前，`new` 失效时会返回零，这和 C 语言中的 `malloc` 失效时返回 `NULL` 指针值的方式非常相似。我们通过发出 `bad_alloc`（错误分配）异常事件显示了新样式的 `new` 失效。我们说明了如何使用 `set_new_handler` 来指定自定义函数，调用这个函数来处理内存耗尽的情形。我们讨论了 `auto_ptr` 类模板，以确保动态分配的内存会被正确删除以避免内存泄漏。

第 24 章“Java 应用程序与小程序入门”介绍了典型的 Java 程序设计环境，并简要介绍了在 Java 程序设计环境中设计应用程序和小程序。一些输入和输出是通过 `JOptionPane` 这种新的图形用户界面（GUI）来执行的，它能够为输入和输出提供预定义的窗口（叫作对话框）。`JOptionPane` 把输出数据发送到窗口中，并从窗口中获得输入数据。本章使用几个与 Java 2 Software Development Kit（J2SDK，Java 2 软件开发工具）一起提供的范例演示小程序来介绍小程序。我们使用 `appletviewer`（与 J2SDK 一起提供的工具）来执行这几个范例小程序。然后，我们编写了一些 Java 小程序，这些小程序所执行的任务与本章前面部分中所编写的应用程序类似。我们还解释了小程序和应用程序之间的相同点和不同点。学完本章之后，学生将知道如何编写出简单而完整的 Java 应用程序和小程序。下面这几章同时使用了小程序和应用程序来演示说明了其他重要的程序设计概念。

第 25 章“Java 的运算符、方法与数组”集中讲述了 Java、C 和 C++之间的共同点和不同点。本章讨论了 Java 中的基本数据类型以及它们是如何与 C/C++区别，同时还讨论了术语中的一些不同之处。例如，我们在 C/C++中称作函数的部分，在 Java 中叫作方法。本章中还包含了对逻辑运算符的讨论，这些逻辑运算符是&&（逻辑与）、&（布尔逻辑与）、||（逻辑或）、|（布尔逻辑同或）、^（布尔逻辑异或）以及！（非）。我们解释了方法重载（与 C++中的函数重载相同）这一主题。在本章中，我们还介绍了事件和事件处理，这些是设计图形用户界面的要素。事件就是诸如按钮单击、鼠标单击、按键盘上的键这样的状态改变的通知。Java 允许程序员通过事件处理程序来响应各种事件。我们还介绍了 Java 中的数组，它被处理为一个完整的对象。这是 Java 承诺 100%面向对象的进一步的证据。我们还讨论了把数据结构化为相同类型的相关数据项的数组或组。本章介绍了大量单下标数组和双下标数组的例子。

第 26 章“基于对象的 Java 编程”更加深入地讨论类。本章集中讨论了类和对象的精髓和术语。对象是什么？对象的类是什么？对象的内部是什么样子？如何创建对象？如何破坏对象？对象如何与其他对象进行通信？对于把软件封装为可重用的组件而言，为什么类是非常自然的机制？本章讨论了抽象数据类型的实现，这些抽象数据类型包括 Java 样式的类、访问类成员、使用 private 变量来增强信息隐藏、从实现过程中分离接口、使用访问方法和工具方法、使用构造函数初始化对象以及使用重载构造函数。本章讨论了声明和使用固定的引用、组合，建立把对对象的引用作为其成员的类的过程、使对象“知道自身”的 this 引用、动态内存分配、用于维护和处理整个类范围内的数据的 static 类成员，以及诸如堆栈和队列这样的流行的抽象数据类型的实例。本章还介绍了 package 语句，并讨论了如何创建可重用的包。本章练习要求学生开发出用于复杂数字、随机数字、时间、日期、矩形和长整型数的类、一个用于玩 Tic-Tac-Toe 的类、一个存款帐号类以及一个用于保存成组整数的类。

第 27 章“面向对象的 Java 编程”讨论了对对象的类和相关类的程序设计之间的关系。我们如何在对象类之间利用通用性，以使建立大型软件系统所花的工作量最小？什么是多态性？“用通用方式编程”而不是“用特殊方式编程”，这意味着什么？按通用方式进行程序设计如何使系统的修改变得简单，以及如何只需最小的工作量就能够添加新功能？我们如何能够为所有对象编程，而不是对每种类型的对象单独进行编程？本章讲解了面向对象编程语言的最基础的功能之一，即继承，它是一种可重用的形式，在这种形式中，通过吸收现有类的功能并添加合适的新功能，就可以快速而容易地开发出新类。本章讨论了超类和子类、protected 成员、直接超类、间接超类、在超类和子类中构造函数的使用，以及带有继承的软件工程的概念。我们还介绍了可帮助隐藏实现细节的内部类。内部类最常用于创建 GUI 事件处理程序。指定的内部类能够在其他类的内部进行声明，它们在为多个 GUI 组件定义通用的事件处理程序方面非常有用。匿名的内部类是在方法内部声明的，它们用于创建对象，通常是用于特定 GUI 组件的事件处理程序。本章把继承（“是一种”关系）与组合（“具有一种”关系）进行了对比。本章的一个特点就是它通过实际案例学习了点、圆和圆柱类层次结构的实现。这些练习要求学生去比较通过继承创建新类和通过组合创建新类；扩展本章中所讨论的继承层次结构；为四边形、

梯形、平行四边形、矩形和正方形编写出继承的层次结构，以及使用二维形状和三维形状创建出更加通用的形状层次结构。本章解释了多态性行为。当很多类通过继承一个通用超类的方式联系在一起时，每个子类对象都可以作为一个超类对象。这使程序能够按照一种独立于特定类型子类对象的通用方式进行编写。新类型的对象能够被相同的程序进行处理，因此使系统具有更强的可扩展性。多态性使程序能够消除复杂的 `switch` 逻辑，支持简单的“直线型”逻辑。例如，视频游戏的屏幕管理程序能够向待绘制的对象链接表中的每个对象发送一个“绘图”信息。每个对象都指定如何绘制自身。只要新对象也知道如何绘制自身，那么无需对该程序进行修改就可以把新类型的对象添加到这个程序中。这种编程样式通常用于实现今天流行的图形用户界面。本章区分了抽象（`abstract`）类（在抽象类中，对象不能初始化）和具体类（在具体类中，对象能够初始化）。本章还介绍了接口，即必须由执行该接口的任何类进行定义的方法组。接口是 Java 中用于代替 C++ 中名为多重继承的危险（尽管功能很强大）功能的组件。

第 28 章和第 29 章是与我们的同事 Tem Nieto 先生合作完成的，他在 Deitel & Associates 公司工作。Tem 无尽的耐心、对细节的关注、说明能力和创造性都在这些章节中得到了淋漓尽致的体现。

第 28 章“Java 图形和 Java2D”开始介绍 Java 的多媒体内容。传统的 C 和 C++ 程序设计都相当局限于字符模式的输入/输出。一些版本的 C++ 得到了一些能够处理图形而且依赖于平台的类库的支持，但使用这些库，会使应用程序不具有可移植性。Java 的图形功能不依赖于平台，因此具有可移植性。您能够开发出图形密集型的 Java 小程序，并在万维网上把它们发布给自己在各地的同事，他们就能够在本地的 Java 平台上很好地运行这些小程序。我们讨论了图形背景和图形对象；绘制字符串、字符和字节；颜色和字体控制；屏幕控制和着色模式；以及绘制线条、矩形、圆角矩形、三维矩形、椭圆形和多边形。我们还介绍了 Java2D API，它是 Java 2 中的新组件，提供了功能强大的图形处理工具。本章有很多图，这些图与实时代码实例、生动的屏幕输出、详细的功能表以及详细的艺术线条一起图解说明了每一个图形功能。

第 29 章“Java GUI 组件”介绍了带有用户友好的图形用户界面（GUI）的小程序和应用程序的创建过程。本章集中讨论了 Java 新的 Swing GUI 组件。这些不依赖于平台的 GUI 组件是完全用 Java 编写的。这为 Swing GUI 组件提供了很大的灵活性，它们能够被定制为看起来与程序执行的计算机平台相似，或者它们也能够使用标准的 Java 外观与感觉，可以提供在所有计算机平台上都相同的用户界面。我们讨论了新的 `javax.swing` 包，它能够提供比 `java 1.1` 中 `Java.awt` 组件的功能强大得多的 GUI 组件。通过很多程序、表格和线条图，本章图解说明了 GUI 的设计原理、`javax.swing` 的层次结构、标签、按钮、文本字段、文本框、组合框、复选框、面板、滚动面板、自定义面板、处理鼠标事件、窗口、菜单以及使用 Java 的 3 个较简单的 GUI 布局管理器，也就是 `FlowLayout`、`BorderLayout` 和 `GridLayout`。本章的重点是 Java 用于 GUI 处理的委托事件模型。这些练习要求学生去创建特定的 GUI，练习各种 GUI 功能，开发出能让用户使用鼠标和控制字体进行绘图的绘图程序。

第 30 章“Java 多媒体：图像、动画和音频”讨论了 Java 中使计算机应用程序“动起来”