

国外经典计算机科学教材

Object-Oriented
Programming in C++ Fourth Edition

C++

面向对象程序设计

(第四版)

[美] Robert Lafore 著
邓子梁 胡勇 译

全球销量超过
20万册的C++教材

SAMS



中国电力出版社

www.infopower.com.cn

国外经典计算机科学教材

Object-Oriented Programming in C++ Fourth Edition

C++

面向对象程序设计

(第四版)

[美] Robert Lafore 著
邓子梁 胡勇 译



中国电力出版社

www.infopower.com.cn

Object-Oriented Programming in C++ , Fourth Edition (ISBN 0-672-32308-7)

Robert Lafore

Authorized translation from the English language edition, entitled Object-Oriented Programming in C++ , Fourth Edition, published by Sams, Copyright©2002

All rights reserved.

No part of this 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.

CHINESE SIMPLIFIED language edition published by China Electric Power Press Copyright©2003

本书由美国培生集团授权出版。

北京市版权局著作权合同登记号 图字：01-2002-4733 号

图书在版编目（CIP）数据

C++面向对象程序设计（第四版） / （美）拉福（Lafore,R.）著；邓子梁等译.

—北京：中国电力出版社，2003

（国外经典计算机科学教材）

ISBN 7-5083-1910-9

I.C... II.①拉...②邓... III.C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字（2003）第 111165 号

丛 书 名：国外经典计算机科学教材

书 名：C++面向对象程序设计（第四版）

编 著：（美）Robert Lafore

翻 译：邓子梁 等

责任编辑：朱恩从

出版发行：中国电力出版社

地址：北京市三里河路6号

邮政编码：100044

电话：（010）88515918

传 真：（010）88518169

印 刷：汇鑫印务有限公司

开 本：787×1092 1/16

印 张：47.25

字 数：1206千字

书 号：ISBN 7-5083-1910-9

版 次：2004 年 2 月北京第 1 版

2004 年 2 月第 1 次印刷

定 价：69.80 元

版权所有 翻印必究

出版说明

新世纪的朝阳刚刚露出丝抹微红，如火如荼的全球信息化浪潮便汹涌而至，让人无时无刻不感受到新一轮产业革命的气息。如何在这场变革中占尽先机，既是对民族信息业的挑战，也是机遇。从而，作为民族信息产业发展基石的高等教育事业就被赋予了比以往更重的责任，对培养和造就我国 21 世纪的一代新人提出了更高的要求。但在计算机科学突飞猛进的同时，专业教材的发展却严重滞后，越来越成为人才培养的瓶颈。同时，以美国为代表的西方国家计算机科学教育经历了充分的发展，产生了一批有着巨大影响力的经典教材，因此，以批判、借鉴的态度有选择地引进这些国外经典计算机教材，将促进国内教学体系和国外接轨，大大推动我国计算机教育事业的发展。

中国电力出版社进入计算机图书市场已有近 6 个年头，通过坚持“高端、精品、经典”战略，致力于与国外著名出版机构合作，出版了大批博得计算机业界和教育界赞誉的作品。通过与信息技术教育界人士的广泛沟通，同时依托丰富的出版资源，中国电力出版社适时推出了“国外经典计算机科学教材”的出版计划。本次教材出版计划是和美国最大的计算机教育出版机构——Pearson 教育集团（Addison-Wesley、Prentice-Hall 等皆为其下属子公司）合作，依托其数十年积累的大批经典教材资源，确保了教材选题的权威经典。

为保证这套教材的含金量，并做到有的放矢，我们在国内组织了由中国科学院、北京大学等一流院校教师组成的专家指导委员会，对高校课程教学体系做了系统、详细的调查，听取了众多教育专家、行业专家的意见，对教育部的教育规划进行了认真研究，并深入了解国外大学实际教学选用的教材状况，对国外教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务，来作为本次“国外经典计算机科学教材”出版计划的宗旨。

我们从 2002 年的下半年开始着手这套教材的策划工作，并多次组织了专家研讨会、座谈会等，分析现有教材的优点与不足，采其精华，并力争体现本套教材的质量和特色。

1. 深入理解国内的教学体系结构，并比照国外相同专业的课程设置，既具有现实的适用性，又立足发展眼光，具备一定的前瞻性。

2. 以计算机专业的核心课程为基础，同时配合专业教学计划，争取覆盖专业选修课程和专业任选课程。

3. 选取国外的最新教材版本，同时对照国内同专业课程的学时要求，对不适用的版本进行剔除，充分满足国内教学要求。

4. 根据专业对口和必须具备同课程教学经验的要求，严格挑选译者，并严把质量关，确保教材翻译的高质量。

5. 通过从原出版社网站下载勘误表及与原书作者进行沟通的方式，对原书中的错误一一做了修改。

6. 对教材出版的后期工作，如审校、编辑、排版、印刷进行了严格的质量把关。

经过专家指导委员会的集体讨论，并广泛听取广大高等院校师生的意见，反复比较，从数百种国外教材中遴选出数十种，列入第一阶段的出版计划。这些教材的作者无一不是学富五车的大师，如 Stallings, Date, Ullman, Aho, Bryant, Sedgewick 等，他们的作品均是一版再版，并被众多国外一流大学如 Stanford University, MIT, UC Bekerley, Carnegie Mellon Univeristy, University of Michigan 等采用为教材。拟订的第一阶段出版计划包括 30 种图书，内容覆盖程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等计算机专业核心基础课程，基本满足国内计算机专业的教学要求。

此外，为了帮助广大任课教师加深对本系列教材的理解，减轻他们的备课难度，我们从国外出版机构引进了大批的课程教学辅助资料，并积极延请国内优秀教师，根据其使用该系列教材中的教学经验，着手编写更加适合国内应用状况的教辅材料。

由于我们对国内高校计算机教育存在认识深度上的不足，在选题、翻译、编辑加工出版等方面的工作中还有许多有待提高之处，恳请广大师生和读者提出批评和建议，并期待有更多的人加入到我们的工作中来。我们的联系方式是：

电子邮件：csbook@cepp.com.cn

联系电话：010-88515918-300

联系地址：北京市西城区三里河路 6 号中国电力出版社

邮政编码：100044

译者序

毫无疑问，C++语言是当今世界上最流行也是最强大的程序设计语言，成千上万的程序员为之疯狂、为之倾倒并追随终身。这种状况已经持续了二十多年，仍未见其衰败迹象，它依然牢牢统治着无数的程序员。我们也许想问，C++的魅力何在？现在，《C++面向对象程序设计》将给你一个完整的回答。

C++的统治地位来源于它对上一个统治者——C 语言的宽容，更为重要的是其自身的强大功能，以及它不断发展和完善的能力。正因为如此，C++的经典作者 Robert Lafore 于近期推出了畅销书《C++面向对象程序设计》的第四版。在这个最新版本中，作者大规模修订了原来的内容，通过全面结合 UML 等面向对象理论，反映了 C++的最新进展——标准 C++和软件开发中的新思想、新概念和新方法。

本书作为较全面的 C++学习书籍，覆盖了从基本概念到完整面向对象应用程序的各方面内容，最为适用于初中级水平的读者。本书的前几版曾将成千上万的读者领入了 C++面向对象程序设计的大门，而本书是它的最新版本——第四版。

通过本书，读者将基本掌握的内容主要包括：C++语言的基本知识、C++的高级特性、面向对象的设计与分析、统一建模语言（UML）和标准模板库等。按照这些内容，本书可以大致分为几个部分，第 1 章至第 7 章主要讲述了 C++的基础知识，如循环、选择、结构、函数、对象、类、数组和字符串等等；第 8 章至第 14 章主要讲述了 C++的高级特性，如重载、继承、虚函数、流、文件、模板和异常等等；第 15 章的标准模板库（STL）是标准 C++新增的解决通用程序设计问题的一种方法；第 16 章讲述了面向对象软件的开发方法。附录主要讲解了在 C++学习中常用的编程环境。

本书的原版已经证明，此书极为适合刚刚涉足编程领域的初学者和已有一定编程经验的中级读者。另外，它所包含的上百个练习也为教师们提供了极大的帮助，这使得它能够成为一本成功的教材，为许多高等学校广泛采用。书中每章后面的习题，都可以用于检测自己的学习效果。所有思考题都有答案，少量练习也附有答案。

本书的翻译工作由邓劲生、胡勇、徐啸及刘建等人组织，虎贲工作室的全体人员参加了翻译、录入、校对和排版工作，全书最后由邓劲生统稿。尽管我们都在 C++编程领域有着打拼多年的丰富经验，但是仍然感觉从书中受益匪浅。不过，由于时间紧迫且译者水平有限，错误之处在所难免，恳请广大读者批评指正。

译者

前 言

《C++面向对象程序设计》第四版的主要变化包括对统一建模语言（UML）的初步介绍，第13章增加的文件间通信部分，以及第16章介绍的用于改进软件开发的方法。

开始介绍 UML 时，只要符合文章主题就应尽量使用 UML 图例，所以全书中有很多新的 UML 图例。文件间通信部分集中了先前分散在全书的很多概念。自本书上一个版本以来，业界关于面向对象分析和设计的方法已有了不少发展，因此本书对与这个主题有关的章节做了修改，从而反映这些最新的进展。

从上一个版本以来，C++本身只有很少改变。虽然如此，除了已提及到的修订部分，还做了很多小的变动，以便更清楚地阐明已存在的主题，另外还改正了不少印刷错误，对原文中不准确的地方也做了修正。

关于作者

Robert Lafore 从 1982 年开始写作计算机编程方面的书籍。他编写的畅销书包括《Assembly Language Programming for the IBM PC》、《C Programming Using Turbo C++》、《C++ Interactive Course》和《Data Structures and Algorithms in Java™》（该书中文版《Java 数据结构和算法》已由中国电力出版社出版）。Lafore 拥有数学和电气工程学位，并且从 PDP-5 时代（那时 4KB 主存还被认为是奢侈的）就开始编程了。他的兴趣包括徒步旅行、帆板运动和趣味数学。

献 词

本书献给 GGL 和她不屈不挠的精神。

第四版致谢

向那些通过电子邮件和我讨论及纠错的读者们致谢。我也感激下面这些给我提供建议和更正的计算机专家们：丹佛里吉斯大学的 Bill Blomberg，南卡罗莱纳州查尔斯顿学院的 Richard Daehler-Wilking，瑞典皇家技术学会的 Frank Hoffmann，加州圣何塞州立大学的 David Blockus。我要特别感谢加州弗里蒙特 Ohlone 学院的 David Topham，感谢他大量详细的构思和对问题的敏锐眼光。

在 Sams 出版公司，Michael Stephens 在出版细节中提供了专业而友好的联络。审稿人 Robin Rowe 和技术编辑 Mark Cashman 为帮助我做了大量努力，如果不成功则完全是我的过错。项目经理 Christina Smith 在令人惊讶的短时间内安排好了每一件事情，Angela Boley 使每一件事都顺利进行，最后，Matt Wynaalda 进行了专门的校对。我要感谢你们所有的人。

第三版致谢

我要感谢 MacMillan 计算机出版公司的整个团队。特别地，Tracy Dunkelberger 在整个项目中发挥了很好的先锋带头作用，在冗长的进度计划中表现出了良好的耐性。Jeff Durham 就无数细节问题熟练而富含幽默感地充当了我和编辑们的沟通渠道。Andrei Kossorouko 以他 C++ 的专家经验保证了我的这个版本做得更好而不是更差。

第二版致谢

向下列教师致谢，他们在其学院或大学里使用这本书作为课本，感谢他们在策划第二版中所做的帮助：Dave Bridges、Frank Cioch、Jack Davidson、Terrence Fries、Jimmie Hattemer、Jack Van Luik、Kieran Mathieson、Bill McCarty、Anita Millsbaugh、Ian Moraes、Jorge Prendes、Steve Silva 和 Edward Wright。

我要感谢对第一版提出更正和建议的大量读者，这些更正和建议中很多都非常有价值。

在 Waite 出版集团公司，Joanne Miller 严格执行了我的进度计划，并且充当了我的学术联络人；Scott Calamar，任何时候都保证每一个人知道他在做什么；Deirdre Greene 作为一名审稿编辑，有着一双锐利的眼睛。

同样感谢 Mike Radtke 和 Harry Henderson，他们为此书做了专业的技术复查。

向西俄勒冈州立学院的 Edward Wright 致以特别的感谢，他复查和试做了新的练习。

第一版致谢

我主要感谢 Mitch Waite，每一张手稿都倾注了他辛勤细致的劳动，此外，他还提供了无数有用的建议。

感谢 Azusa Pacific 大学的 Bill McCarty，他复审了手稿的内容以及内容的课堂适应性，并做了大量出色的改进，还改正了我晦涩的拼写。

感谢 George Leach 运行了所有的程序，并且还令人震惊地发现了几个在某些情况下不能正确演示的程序。我相信这些问题都已经被改正；如果没有，那么这些错误的责任全部在我。

感谢 Waite 集团的 Scott Calamar 为写作和出版这本书处理了无数组织方面的问题。他的才能和无穷尽的幽默是这本书能够按时完成的重要因素。

我还要感谢 Borland 的 Nan Borreson 所提供的最新软件版本，Harry Henderson 复查了所有的练习，Waite 集团的 Louise Orlando 使本书顺利进入了生产过程，Matrix Production 的 Merrill Peterson 协调了大量生产过程中的问题，Juan Vargas 做了创新性的设计，最后，Frances Hasegawa 以她不可思议的才能重绘了我的草图并使其成为了漂亮而醒目的艺术品。

内容简介

这本书将讲解如何使用 C++ 程序设计语言来编写程序。然而，它做得比这更多。在过去的几年内，软件开发领域出现了一些重大的革新。本书将根据这些最新的发展来教授 C++。现在来看看它们都是什么吧！

程序设计变革

在 20 多年以前，程序员在接到一个项目后可以立即坐下来编代码。然而，随着程序设计项目变得越来越大和越来越复杂，这种方法就不怎么有效了。问题的根源就是复杂性。

大型程序是人类所创建过的最复杂的实体。由于它的复杂性，程序很容易出错，软件错误往往代价高昂，有时甚至威胁到生命安全（如空中交通控制系统）。为解决软件的复杂性问题，程序设计领域出现了三个主要的变革，它们是：

- 面向对象程序设计（Object-Oriented Programming, OOP）
- 统一建模语言（Unified Modeling Language, UML）
- 改良的软件开发过程

本书将用这三个变革的思想来讲授 C++ 语言。在此，你不仅会学到一门计算机语言，还会学到很多有关软件开发的新方法。

面向对象程序设计

为什么面向对象程序设计成为大多数软件项目的首选方法？OOP 提供了一种全新的强大方法来处理复杂性。相对于将程序看成一系列可执行的步骤，新的方法将程序看成是一组具有特定属性和执行特定操作的对象。在进一步学习该方法之前，它听起来有些晦涩，但它确实能够让程序更清晰、更可靠和更容易维护。

本书的一个主要目的就是讲授面向对象程序设计，因此将尽可能早地介绍它，并且尽可能覆盖它的所有主要特性。本书的大多数示例程序都是面向对象的。

统一建模语言

统一建模语言（UML）是一门由多种图例组成的图形语言。它能够帮助程序分析员弄清程序应该做什么，同时帮助程序员设计和理解程序的工作过程。UML 是一种能使程序设计更为简单和有效的强大工具。

在第 1 章中对 UML 做了一个概述，指出本书中将讨论的 UML 的具体特性。在有助于阐明所讨论的 OOP 话题的地方，都介绍了相关的 UML 特性，这种方法使得在学习 UML 的同时也有助于学习 C++。

语言和开发平台

在面向对象程序设计语言中，C++ 是到目前为止使用最为广泛的语言。最近才加入 OO 语言

113549/02

一族的 Java 由于缺少某些特性，如指针、模板和多重继承等，其功能没有 C++ 多，也不如 C++ 强大（如果你想学习 Java，C++ 的学习将是一个起点，因为 Java 语法和 C++ 很相似）。

最近还出现了其他几种 OO 语言，比如 C#，但它们还没有像 C++ 一样被广泛接受。

直到最近，C++ 标准还处在不断变革中，这意味着每一个编译器厂商在处理某些细节时都会有所不同。然而，在 1997 年 11 月，ANSI/ISO C++ 标准委员会批准了最后一个 C++ 草案，这就是众所周知的标准 C++（ANSI 是指美国国家标准化学会，ISO 是指国际标准化组织）。标准 C++ 增加了许多新的特性，如标准模板库（Standard Template Library, STL）。本书以标准 C++ 为主（少数地方除外，会在学习中标明）。

最流行的 C++ 开发环境由 Microsoft 和 Borland 提供，它们运行在各种 Microsoft Windows 平台上。我们尽量保证本书所有的示例程序都可以运行在 Borland 和 Microsoft 编译器的当前版本下（参见附录 C “Microsoft Visual C++” 和附录 D “Borland C++ Builder” 中更多关于编译器的内容）。

关于本书

本书使用 C++ 程序设计语言来讲授面向对象程序设计，具体的编译器可以是 Microsoft 的，也可以是 Borland 的。此外，本书还介绍了 UML 和软件开发过程。本书适合于专业程序员、学生和程序爱好者。

新概念

对于传统语言（如 Pascal、Basic 和 C）程序员来说，OOP 涉及的概念是很新的。类、继承和多态性等思想是 OOP 的核心概念，但是在讨论面向对象语言的细节时，很容易忘掉这些概念。许多书籍都过多地让读者关注语言特性的细节，而忽略掉了这些特性存在的缘由。而本书着眼于全局，将具体的细节和宏观的概念相联系。

渐进式方法

本书采用了渐进式的学习方法，从最简单的程序设计示例开始，到最后完整的面向对象应用程序。我们尽量以适当的速度介绍新概念，以便于读者有足够的时间在下一次学习之前消化掉这些思想。我们也尽可能地使用图表来阐明新思想。在每一章的结尾都提供了思考题和程序设计练习题以适合于课堂教学。附录 G 附有思考题和小部分（带星号）练习题的解答，不同难度的练习题可以给学生以不同的挑战。

使用本书的基础

即使读者没有任何的程序设计经验，也可以使用本书。当然，如果有些经验（例如 Visual Basic 的），也不会有坏处。

使用本书不必知道 C 语言。许多 C++ 的书都假定读者已经知道了 C，但是本书没有。本书从最基本的地方开始讲授 C++。当然，知道 C 也没有什么坏处，但读者会很惊奇在 C 和 C++ 之间只有很少的重叠部分。

读者应该熟悉 Microsoft Windows 的基本操作，如运行应用程序和复制文件。

软件和硬件

首先需要有一个 C++ 编译器。本书的程序都已经通过 Microsoft Visual C++ 和 Borland C++ Builder 测试了。两种编译器都有适合于学生的“学习版”发售。

附录 C 提供了如何操作 Microsoft Visual C++ 的详细信息，而附录 D 介绍的是 Borland 的产品。其他的编译器，如果它们支持标准 C++，也可以运行本书的大部分程序。

其次，计算机必须有足够的处理器速度、内存和磁盘空间，以运行所选择的编译器，具体要求可以参见编译器生产厂商的说明书。

控制台模式的程序

书中有大量的程序示例。它们是控制台模式的程序，可以运行在编译器环境的字符模式窗口下或者直接运行在 MS-DOS 窗口下。这样做可以避免完整 Windows 图形程序的复杂性。

示例程序程序清单

可以到 Sams 出版公司的网站获得本书示例程序的程序清单，该公司网址是：

<http://www.sampublishing.com>

在网站上可以通过本书的 ISBN 号（可在书的前面找到）或者书名搜索到本书的相关数据，然后再单击“Source Code”下载书的示例程序。

简单控制台图形

少数示例程序会用到一个被称为简单控制台图形的图形库来绘制图形。这些图形由控制台字符构成，因此它们不是很复杂，但却可以编写一些有趣的程序。这个库的文件由出版公司的网站连同示例程序清单一起提供。

为编译和运行这些图形示例，需要在程序中包含头文件 MSOFTCON.H 或 BORLACON.H，具体情况根据编译器而定。此外，还需要将 MSOFTCON.CPP 或 BORLACON.CPP 加入到图形示例的项目中。附录 E “简单控制台图形”提供了这些文件的程序清单和使用方法。附录 C 和附录 D 解释了如何在特定的编译器环境中使用文件和项目。

程序设计练习

每章都包含了 12 个左右的练习，每个练习都要求创建一个完整的 C++ 程序。附录 G 提供了每章前三个或四个练习题的答案。余下的练习题，读者可以自行解答（然而，如果你正讲授 C++ 课程，请注意阅读下面的“给教师的提示”）。

比想像的容易

你可能听说过 C++ 很难学，但实际上它和其他语言相似，只不过增加了两、三个“极重要的思想”。这些新的思想本身就很有趣，学习它们是一件非常有意思的事情。另外，它们也正在成为程序设计文化中的一部分，每个人都应多少知道一点，如发展史和心理分析。我们希望本书在帮助你学习这些新思想的同时，讲授 C++ 程序设计的细节技术。

给教师的提示

教师及其他已经知道 C++ 和 C 的人，将会对本书所使用方法的细节和组织方式感兴趣。

标准 C++

书中的所有程序都兼容于标准 C++，很少的例外也是由于需要将就编译器的特殊习惯。我们花了一章的篇幅来讲授标准 C++ 中的标准模板库（STL）。

统一建模语言（UML）

在上一个版本中，是在最后一章介绍 UML 的。但在当前版本中，已将 UML 集成进了整本书，哪里有需要，哪里就介绍 UML。例如，在第一次演示不同类之间的通信时介绍了 UML 类图，而在“继承”一章中介绍了泛化关系。

第 1 章“概述”包含了本书所介绍的 UML 内容的程序清单。

软件开发过程

正式的开发过程正成为程序设计中日益重要的一个方面。另外，学生们对面向对象程序的设计过程也时常感到很迷惑。因为这些原因，本书用了一章专门讲述软件开发过程，强调了面向对象的程序设计方法。在上一个版本中，我们集中讲述了 CRC 卡片，但软件开发已经将重点移向了场景分析法，所以本书将使用后者来分析程序项目。

C++ 和 C 不同

少数学院要求他们的学生在学习 C++ 前应首先学习 C。我们认为这是错误的，C 和 C++ 是完全不同的两种语言。虽然它们的语法相似且 C 是 C++ 的一个子集，但这种相似性只是一个历史的偶然。实际上，C++ 程序的基本方法和 C 程序的基本方法有着本质的区别。

C++ 已经取代了 C 成为大型程序开发的首选语言，因此不必迷信地认为在讲授 C++ 前讲授 C 是必需的。不懂得 C 的学生，反而节省了先学 C 后学 C++ 中的麻烦，因为这是一种很没效率的方法。那些已经懂得了 C 语言的学生可以跳过某些章节的内容，但他们会发现相当大部分的内容是新的。

根据 OOP 优化本书结构

我们将从 C 和 C++ 共有的过程化概念开始讲授，消化完这些概念后就转向新的 OOP 概念。这看起来好像是反其道而行，但由于目标是尽可能快地开始真正的面向对象程序设计，因此只是在讲述第 6 章的类之前提供了少量的面向过程基础。即使在开始的几章，也是充满了 C++ 的用法，而不是 C 语言的。

比起传统的 C 语言书籍，本书有些内容介绍得较早。例如，结构是理解 C++ 的一个关键特性，因为在句法上，类就是结构的一个扩展版本。由于这个原因，在第 5 章中介绍了结构类型，以便在讨论类时可以对概念熟悉一点。

某些概念（如指针），在传统 C 语言书籍中一般放在书的后部分讲解。因为介绍 OOP 概念并不需要懂得指针，而且指针通常是 C 和 C++ 学习中的绊脚石，因此将对指针的讨论推迟到了主要的 OOP 概念都已经消化之后。

使用 C++ 高级特性作替换

某些 C 的特性已经由 C++ 的新方法替换了。例如，printf() 和 scanf() 函数是 C 中的主力输入输出函数，但在 C++ 中已很少使用了，因为 cout 和 cin 可以做得更好。因此书中省略了对这些函

数的描述。类似地，C 中的 `#define` 常量和宏也由 C++ 中的 `const` 指示符和内联函数代替，所以在此只做简要的介绍。

缩减不相关特性

由于本书的重点是面向对象程序设计，所以某些 C 语言中不常使用的或者与 OOP 没有特别关系的特性就被省略了。例如，理解 C 语言中的位操作运算符（用来操作单个的位）就与学习面向对象程序设计没有太大的联系。所以在讨论中就将这些特性省略或只做简要介绍，这对理解 C++ 的主要特性没有任何损失。

本书的重点集中在 OOP 的基础技术，着重于轻松带领读者理解新的概念，再将它们应用于实际的程序设计问题中。

程序设计练习

书中没带星号的练习题没有答案，但是，负责的教师可以从 Sams 出版公司的网站上获得解决方案的建议。输入 ISBN 号或者书名可以找到相关数据，然后再下载。

练习题有不同的难度等级。每一章中，前面的练习题相当容易，越往后越有挑战性。教师可以根据班级的水平指定相应难度等级的练习题。

目 录

出版说明	
译者序	
前言	
关于作者	
献词	
内容简介	

第 1 章 概述	1
1.1 面向对象程序设计诞生的原因	1
1.2 面向对象语言的特征	5
1.3 C++和 C	9
1.4 基础知识	10
1.5 统一建模语言 (UML)	10
1.6 本章摘要	11
1.7 思考题	12
第 2 章 C++编程基础	14
2.1 准备开始	14
2.2 基本程序结构	15
2.3 使用 cout 输出	17
2.4 指令	18
2.5 注释	19
2.6 整数变量	20
2.7 字符变量	24
2.8 使用 cin 输入	26
2.9 浮点数类型	28
2.10 bool 类型	30
2.11 setw 操作符	31
2.12 变量类型总结	33
2.13 类型转换	34
2.14 算术运算符	38
2.15 库函数	41

2.16	本章摘要	44
2.17	思考题	44
2.18	练习	46
第 3 章	循环和选择	48
3.1	关系运算符	48
3.2	循环	50
3.3	选择	61
3.4	逻辑运算符	77
3.5	优先级总结	80
3.6	其他控制语句	81
3.7	本章摘要	84
3.8	思考题	85
3.9	练习	86
第 4 章	结构	89
4.1	结构	89
4.2	枚举	101
4.3	本章摘要	107
4.4	思考题	107
4.5	练习	108
第 5 章	函数	111
5.1	简单的函数	112
5.2	向函数传递参数	115
5.3	从函数返回值	122
5.4	引用参数	127
5.5	重载函数	132
5.6	递归	135
5.7	内联函数	137
5.8	默认参数	138
5.9	域解析运算符和存储类	140
5.10	通过引用返回	144
5.11	常量类型的函数参数	146
5.12	本章摘要	147
5.13	思考题	147

5.14 练习	149
第 6 章 对象和类	151
6.1 简单的类	151
6.2 作为现实对象的 C++对象	157
6.3 作为数据类型的 C++对象	159
6.4 构造函数	160
6.5 作为函数参数的对象	164
6.6 默认拷贝构造函数	168
6.7 函数返回对象	170
6.8 纸牌游戏的例子	172
6.9 结构和类	175
6.10 类、对象和内存	176
6.11 静态类成员数据	177
6.12 const 和类	179
6.13 类和对象的意义	182
6.14 本章摘要	183
6.15 思考题	183
6.16 练习	185
第 7 章 数组和字符串	188
7.1 数组基础	188
7.2 作为类成员数据的数组	200
7.3 对象数组	203
7.4 C 字符串	208
7.5 标准的 C++ string 类	218
7.6 本章摘要	225
7.7 思考题	225
7.8 练习	227
第 8 章 运算符重载	230
8.1 重载一元运算符	231
8.2 重载二元运算符	237
8.3 数据转换	249
8.4 UML 类图	260
8.5 运算符重载与转换中的陷阱	261

8.6	关键字 <code>explicit</code> 和 <code>mutable</code>	262
8.7	本章摘要	265
8.8	思考题	265
8.9	练习	267
第 9 章	继承	270
9.1	派生类和基类	271
9.2	派生类构造函数	276
9.3	重载成员函数	278
9.4	使用了哪个函数	279
9.5	英制 <code>Distance</code> 类中的继承	280
9.6	类的层次	283
9.7	继承与图形	286
9.8	公共的和私有的继承	289
9.9	继承的级别	291
9.10	多重继承	294
9.11	<code>EMPMULT</code> 中的 <code>private</code> 派生	299
9.12	多重继承的多义性	302
9.13	聚合关系：类内部的类	304
9.14	继承和程序开发	308
9.15	本章摘要	309
9.16	思考题	309
9.17	练习	311
第 10 章	指针	314
10.1	地址和指针	315
10.2	取地址运算符 <code>&</code>	315
10.3	指针和数组	322
10.4	指针和函数	325
10.5	指针和 C 类型字符串	331
10.6	内存管理： <code>new</code> 和 <code>delete</code>	336
10.7	对象指针	340
10.8	链表实例	344
10.9	指向指针的指针	348
10.10	解析算术表达式实例	352
10.11	模拟赛马	356