

双语教材 + 多媒体教室

C++大学教程 (第二版)

C++ How to Program
Second Edition

[美] Harvey M. Deitel 著
Paul James Deitel

邱仲潘 等译
冯 平 审校



电子工业出版社

Publishing House of Electronics Industry
www.phei.com.cn



国外计算机科学教材系列

C++ 大学教程

(第二版)

C++ How to Program

Second Edition

[美] Harvey M. Deitel 著
Paul James Deitel

邱仲潘 等译

冯 平 审校

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书的作者Deitel一家是美国编程教材的名家,他们的作品繁多,且多为美国各著名大学的指定教材。本书是一本C++编程方面的优秀教程,全面地介绍了过程式编程与面向对象编程的原理和方法,细致地分析了各种性能问题、移植性问题和可能出错的地方,介绍了如何提高软件工程质量,详细讨论了新的ANSI C++标准和标准模板库(STL),并提供了丰富的自测练习和习题。可以说本书是最好的学习C++语言的教程之一,是学习C++编程的“宝典”。本书所配的“多媒体教室”光盘更是独具特色,通过交互式环境,使学习过程更加生动有趣。本书是高等院校进行编程语言和C++教学的教材,是软件设计人员进行C++程序开发的宝贵参考资料,是一本值得珍藏的好书。

Authorized translation from the English language edition published by Prentice-Hall, Inc. Copyright © 1998.

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.

Simplified Chinese language edition published by Publishing House of Electronics Industry. Copyright © 2001.

本书中文简体专有翻译出版版权由Pearson教育集团所属的Prentice-Hall, Inc.授予电子工业出版社。其原文版权及中文翻译出版版权受法律保护。未经许可,不得以任何形式或手段复制或抄袭本书内容。

图书在版编目(CIP)数据

C++大学教程(第二版)/(美)戴特(Deitel, H. M.)等著;邱仲潘等译.

—北京:电子工业出版社,2001.7

国外计算机科学教材系列

书名原文:C++ How to Program, Second Edition

ISBN 7-5053-6727-7

I. C... II. ①戴... ②邱... III. C语言-程序设计-教材 IV. TP312

中国版本图书馆CIP数据核字(2001)第041030号

丛 书 名: 国外计算机科学教材系列

书 名: C++大学教程(第二版)

原 书 名: C++ How to Program, Second Edition

著 者: [美] Harvey M. Deitel Paul James Deitel

译 者: 邱仲潘 等译

审 校 者: 冯 平

责任编辑: 马 岚 冯小贝

排版制作: 今日电子公司制作部

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 53.25 字数: 1363千字

版 次: 2003年3月第5次印刷

书 号: ISBN 7-5053-6727-7
TP·3759

定 价: 78.00元(含光盘一张)

版权贸易合同登记号 图字:01-2000-2079

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

随着 21 世纪的到来, 计算机技术的发展更加迅猛, 在各行各业的应用更加广泛, 越来越多的高等院校增设了有关计算机科学的课程内容, 或对现有计算机课程设置进行了适当调整, 以紧跟前沿技术。在这个教学体系和学科结构变革的大环境下, 对适合不同院系、不同专业、不同层次的教材的需求量与日俱增。此时, 如果能够借鉴、学习国外一流大学的先进教学体系, 引进具有先进性、实用性和权威性的国外一流大学计算机教材, 汲取其精华, 必能更好地促进中国高等院校教学的全局改革。

美国 Prentice Hall 出版公司是享誉世界的高校教材出版商, 自 1913 年成立以来, 一直致力于教材的出版, 所出版的计算机教材为美国众多大学采用, 其中有不少是专业领域中的经典名著, 已翻译成多种文字在世界各地的大学中使用, 成为全人类的共同财富。许多蜚声世界的教授、学者都是该公司的资深作者, 如道格拉斯·科默 (Douglas E. Comer)、威廉·斯大林 (William Stallings) 等。早在 1997 年, 电子工业出版社就从 Prentice Hall 引进了一套计算机英文版专业教材, 并将其翻译出版, 同时定名为《国外计算机科学教材系列》(下称: 第一轮教材)。截至 2000 年 12 月, 该系列教材已出版 23 种, 深受读者欢迎, 被许多大学选为高年级学生和研究生教材或参考书。

4 年过去了, 已出版的教材中多数已经有了后续版本。因此, 我们开始设计新一轮教材 (第二轮教材) 的出版, 成立了由我国计算机界著名专家和教授组成的“教材出版委员会”, 并结合第一轮教材的使用情况和师生反馈意见, 组织了第二轮《国外计算机科学教材系列》出版工作。

第二轮教材的出版原则为:

1. 引进 Prentice Hall 出版公司 2000 年和 2001 年推出的新版教材, 作为替换版本。
2. 在著名高校教授的建议下, 除了从 Prentice Hall 新选了一些教材之外, 还从 McGraw-Hill 和 Addison Wesley Longman 等著名专业教材出版社、麻省理工学院出版社和剑桥大学出版社等著名大学出版社引进了一些经典教材, 作为增补版本。
3. 对于第一轮中无新版本的优秀教材, 我们将其作为沿用版本, 直接进入第二轮使用。
4. 对于第一轮中翻译质量较好且无新版本的教材, 我们将其进行了修订, 也作为沿用版本, 进入第二轮使用。

这次推出的教材覆盖学科范围广、领域宽、层次多, 既有本科专业课程教材, 也有研究生课程教材, 以适应不同院系、不同专业、不同层次的师生对教材的需求。广大师生可自由选择和自由组合使用。

按照计划, 本轮教材规划出版 37 种, 其中替换版本 8 种, 新增版本 14 种, 沿用版本 15 种。教材内容涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。本轮教材计划于 2001 年 7 月前全部出版。教材的使用年限平均为 3 年。我们还将陆续推出一些教材的参考课件, 希望能为授课老师提供帮助。

为了保证本轮教材的选题质量和翻译质量, 我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通

大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本轮教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师和博士,也有积累了几十年教学经验的教授和博士生导师。

在本轮教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括:

1. 对于新选题和新版本进行了全面论证。
2. 对于沿用版本,认真审查了前一版本教材,修改了其中的印刷错误。
3. 对于译者和编辑的选择,达到了专业对口。
4. 对于从英文原书中发现的错误,我们通过与作者联络、从网上下载勘误表等方式,一一做了修改。
5. 对于翻译、审校、编辑、排版、印刷质量进行了严格的审查把关。

通过这些工作,保证了本轮教材的质量较前一轮有明显的提高。相信读者一定能够从字里行间体会到我们的这些努力。

今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

由于我们对国际计算机科学、我国高校计算机教育的发展存在认识上的不足,在选题、翻译、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。

电子工业出版社
2001 年春

教材出版委员会

主 任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委 员	王 珊	中国人民大学信息学院院长、教授
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授 中国计算机学会多媒体专业委员会主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹 鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础教学课程指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

关于作者

Harvey M. Deitel 博士, Deitel & Associates 公司的总裁, 有着近 40 年计算机领域的工作经验, 进行过大量深入的教学研究, 是世界一流的计算机科学教员和研讨会演讲人。Deitel 博士持有麻省理工学院 (MIT) 学士、硕士学位和波士顿大学博士学位。他曾经在 IBM 和 MIT 的虚拟内存操作系统项目中从事研究工作, 开发如今在 UNIX®、Windows NT® 和 OS/2 等系统中广泛实现的技术。Deitel 博士有 20 年的大学教学经验, 取得过“教育特别津贴”, 在和 Paul J. Deitel 一起创办 Deitel & Associates 公司之前, 他是波士顿大学计算机科学系主任。Deitel 博士是几十本专著和多媒体软件包的作者或合著者, 目前正在推出另外五本。Deitel 博士的作品已被翻译成日文、俄文、西班牙文、中文简体、中文繁体、韩文、法文等语言, 他是全球知名的计算机教材作者。

Paul J. Deitel, Deitel & Associates 公司的执行副总裁, 是麻省理工学院 Sloan 管理学校的毕业生, 主修信息技术。通过 Deitel & Associates 公司, 他已经为 Digital Equipment Corporation、Sun Microsystems、White Sands Missile Range、Rogue Wave Software、Software 2000、Computervision、Stratus、Fidelity、Cambridge Technology Partners、Open Environment Corporation、One Wave、Hyperion Software、Lucent Technologies、Adra Systems、Entergy、CableData Systems、NASA at the Kennedy Space Center、the National Severe Storm Center 以及 IBM 等多家公司的客户提供 Java、C 和 C++ 课程的教学活动。Paul 曾为计算机协会波士顿分会讲授 C++ 和 Java 语言, 通过 Deitel & Associates、Prentice Hall 公司和 Technology Education Network 的合作投资讲授卫星直播的 Java 课程。他和 Harvey M. Deitel 博士一起创作了 10 余本专著和多媒体软件包, 目前正在推出另外五本。

Deitel 一家是世界上最畅销的大学计算机科学教程《C How To Program》、《C++ How To Program》、《Java How To Program》的合著者, 也是 Prentice Hall 的第一本多媒体教材《C & C++ Multimedia Cyber Classroom》(即本书所配光盘的前一版) 光盘和《Java Multimedia Cyber Classroom》光盘的合作开发者。这些教材和光盘均已推出第二版。

关于译者

邱仲潘是国内多产的计算机图书翻译作家, 其工作小组还包括张荣、李青、钟铨光、王凌飞、李崇秋、刘云昌、刘昌和、严明英、赖华龙、陈凌峰、陈纯颖、周阳生、邹能东、李耀平、彭振庆、刘文琼、温连英等。

前 言

欢迎使用 ANSI/ISO Draft Standard C++ 语言。本书是由一老一少写成的。老者（HMD，1967 年毕业于麻省理工学院）有着 36 年的编程和教学经验，少者（PJD，1991 年毕业于麻省理工学院）有 15 年的编程经验，并有着丰富的教学与写作经历。老者经验丰富，少者精力充沛。老者追求清晰，少者追求性能。老者喜欢优雅与美观，少者喜欢快速得到结果。一老一少的合作，使本书的信息更丰富、内容更有趣。

这是 C++ 编程者应当高兴和兴奋的时刻，ANSI/ISO C++ 草案标准即将正式应用。ANSI（American National Standards Institute，美国国家标准委员会）和 ISO（International Standards Organization，国际标准化组织）合作开发了这个标准的草案，它即将成为计算机界最重要的全球性标准。

当我们编写《C++ 大学教程》第一版时，我们的目标是为高等学校学生编写一本教材，当时的学生主要学习 Pascal 和 C 语言，强调“过程式”编程模式。向计算机专业一年级和二年级学生提供一本 C++ 教程并不容易，我们必须同时介绍两种编程模式，即过程式编程（因为 C++ 仍然包括 C 语言）和面向对象编程，因此就要在入门介绍中提供两方面的材料。我们在本书的前五章介绍原始数据类型、控制结构、函数、数组、指针和字符串等 C 语言知识，而在第 6 章到第 15 章介绍面向对象编程。

《C++ 大学教程》的第一版已成为世界上使用最普及的大学 C++ 教材。而该书的第二版推迟编写的原因有两个：

1. 这段时间 C++ 正在活跃地发展，新的草案标准文档不断推出，而短期内标准委员会没有接受草案标准的明显迹象。估计还要一年左右标准委员会才能在全球范围内接受当前草案标准。
2. 我们等待着 C++ 大学教程再版的合适时机。1997 年 7 月，Bjarne Stroustrup 推出了《The C++ Programming Language》第三版，Bjarne Stroustrup 是 C++ 的创始人，他的书是 C++ 语言的权威著作。这时我们认为 C++ 的新定义已经相当稳定，可以推出 C++ 大学教程的第二版。

有一段时间，我们把精力转向推出 5 本 Java 方面的书籍：《Java How to Program》第一版与第二版，Windows 95/NT 和 Solaris 平台上的交互式学习软件包《Java Multimedia Cyber Classroom》第一版与第二版，以及《Java How to Program with an Introduction to Visual J++》第一版（由 Microsoft 公司资助）。但 ANSI/ISO C++ 草案标准的即将推出又使我们把精力转向 C++。

《C++ 大学教程》第二版

下面想谈谈本书的创作过程。Prentice Hall 公司的计算机图书编辑 Laura Steele 邀请了 28 位评审人，包括 ANSI/ISO C++ 委员会主席和几位要员，他们中有十几位参加了本书第一版审校并提出了改进建议，特别是设法使本书与 ANSI/ISO C++ 草案标准的演变同步。在此过程中，总共产生了一千多页的审稿材料。然后我们根据 ANSI/ISO C++ 草案标准中增加的部分，修改并编写了新内容。我们编写了代码实例，运行程序以捕获其屏幕输出，并编写了相应的编程步骤。这些内容分布在第一版本的 18 个章节中。然后我们编写了另外三章：第 19 章“string 类与字符串流处理”，第 20 章“标准模板库（STL）”，第 21 章“ANSI/ISO C++ 标准语言补充”。这些章节已经过专家们审读，而且我们认真采纳了他们的建议。例如关于 STL 一章，ANSI/ISO C++ 委员会的多位会员已审阅过该章，并请 STL 的创建者 Alexander Stepanov 和 Meng Lee 进行了一些修改。到本书编写时，C++ 还在变化

之中，只有很少的编辑器能够完全赶上 ANSI/ISO C++ 新标准的速度。在使用 C++ 当前版本与草案标准的 C++ 新特性之间，我们经过了反复地推敲。

软件开发的革命

几年来计算机硬件大大改进，而软件则因为某种原因好像拒绝追求“更快更好”。如今我们正处于软件开发的变革之中，这种变革基于常识、硬件标准化、组件交换性等 Henry Ford 在 Model T Ford 时代采用的思想。这种软件组件称为“对象”或“类”，是产生对象的“模具”。

最早成熟的面向对象语言是 Smalltalk，1970 年初由 Xerox 公司的 Palo Alto 研究中心开发。但最广泛使用的面向对象语言却是 C++ 语言，后者是 20 世纪 80 年代初由 AT&T 的 Bjarne Stroustrup 等人开发的。在本书第一版和第二版之间，出现了另一个重要的面向对象语言，即 Java 语言，是 20 世纪 90 年代初由 Sun 系统公司的 James Gosling 等人开发的。

为什么每 10 年便会推出一种新的面向对象语言呢？Smalltalk 非常超前，是个试验性产品。C++ 正逢其时，适合今天的高性能系统编辑和应用程序开发需求。Java 则提供了建立可移植性、多媒体支持和基于 Internet/World Wide Web 的网络应用。

过程式编程、基于对象编程、面向对象编程与常规编程

本书要介绍 C++ 的五大组件和目前的四种编程机制（下面分五点介绍）。

1. C 语言的过程式编程——第 1~5 章和第 16~18 章，主要课题包括数据类型、控制结构、函数、数组、指针、字符串、结构、位操作、字符操作、预处理等。
2. C++ 过程式编程对 C 语言的改进——3.15 节到 3.21 节，主要课题包括内联函数、引用、默认参数、函数重载和函数模板。
3. C++ 基于对象编程——第 6~8 章，主要课题包括抽象数据类型、类、对象、封装、信息隐藏、成员访问控制、构造函数、析构函数、软件复用性、常量对象和成员函数，复合、友元关系、动态内存分配、static 成员、this 指针等等。
4. C++ 面向对象式编程——第 9~15 章、第 19 章、第 21 章，主要课题包括基类、单一继承、派生类、多重继承、虚函数、动态关联、多态、纯虚函数、抽象类、具体类、流输入/输出、类模板、异常处理、文件处理、数据结构、字符串作为成熟对象、bool 数据类型、强制类型转换运算符、名字空间、运行时类型信息（RTTI）、explicit 构造函数和 mutable 成员。
5. C++ 常规编程——第 20 章，也是本书内容最多的一章，主要课题包括标准模板库（STL）、模板化容器、顺序容器、关联容器、容器适配器、遍历模板化容器的迭代器和处理模板化容器元素的算法。

Pascal 与 C 语言向 C++ 的演变

C++ 已经代替了 C 语言，成为计算机行业系统实现的首选语言。但 C 语言编程依然很重要，是今后十年的重要技术，因为还有许多 C 语言遗留代码需要维护。Harvey M. Deitel 博士二十年来在高等学校从事编程导论课程教学，一直要求编写结构合理、简洁流畅的程序。这些课程中主要介绍基本编程原理，强调有效使用控制结构和功能化。我们按照 HMD 的教学方法提供这些材料。有一些容易出错的地方，我们会及时指出，并说明如何避免。根据我们的经验，学生掌握 C++ 的方式与掌握 Pascal 和 C 语言导论课程的方式差不多，但有一个重要区别，那就是学生们兴趣更大，他们知道

自己是在学习一流的语言 (C++) 和一流的编程机制 (面向对象编程), 而且大学毕业后马上就能使用这种语言。学生的学习热情将会增加, 这一点是非常重要的, 因为 C++ 要比 Pascal 和 C 语言难得多。

我们的目标很明确: 编写一本大学计算机专业的 C++ 编程教科书, 让还没有任何编程经验的学生掌握理论和实践方面要求都很高的 C++ 语言。为了达到这个目的, 本书比其他 C++ 书籍都要厚, 我们不仅介绍 C++, 还要耐心介绍过程式编程、基于对象编程、面向对象编程和常规编程的原理。全世界已经有成千上万的大学生和专业研讨班学员使用这本教材。

从第 1 章就开始介绍面向对象编程

本书的结构曾让我们费尽思量, 是介绍单纯的面向对象编程方法, 还是介绍平衡过程式编程与面向对象编程的混合方法?

许多使用这本教材的老师都曾经进行过程式编程 (Pascal 或 C 语言) 的教学, 而 C++ 本身也不是单纯的面向对象式语言, 而是同时支持过程式编程与面向对象编程。

因此, 我们选择下面的方法。本书前五章介绍 C++ 过程式编程, 讨论计算机概念、控制结构、函数、数组、指针和字符串。这些内容介绍 C++ 中的 C 语言组件和 C++ 对 C 语言的“过程性改进”。

我们通过某种方法使这五章更有特色。每一章的末尾都有一个特殊小节“有关对象的思考”, 这些小节介绍面向对象的术语和概念, 帮助学生熟悉对象的内容和行为。

第 1 章的“有关对象的思考”介绍面向对象的术语和概念。第 2 章到第 5 章的“有关对象的思考”介绍真正的面向对象系统项目的要求, 通过建立一个电梯模拟程序帮助学习了解面向对象设计过程的典型阶段。这些小节介绍如何确定问题中的对象, 如何指定对象的属性与行为, 如何指定对象之间的交互。学完第 5 章后, 学生就可以完成电梯模拟程序的面向对象设计过程, 使用 C++ 编写电梯模拟程序。第 6 章和第 7 章介绍数据抽象与类, 其中“有关对象的思考”介绍了用 C++ 编写电梯模拟程序的各个阶段。

关于《C++ 大学教程》

《C++ 大学教程》一书包含了大量的可应用在各个领域中的程序实例、练习和项目, 使学生可以解决各种有趣的实际问题。本书注重良好的软件工程原则并强调程序的清晰性。我们避开繁琐的语法和术语, 并设法通过例子进行教学。

本书是由两位教育工作者编写的, 这两位都在用大部分时间从事一流实用编程语言的教学工作和教材的创作。

本书的内容特别注重教学方法。例如, 几乎每个文中解释的 C++ 或面向对象编程的新概念都用 C++ 程序实现, 并立即用一个窗口显示程序输出。阅读这些程序就像亲临计算机前输入和运行程序一样, 我们称其为“有生命力的代码”。

文中的教学工具包括每章开头的“教学目标”, 贯穿全文的“常见编程错误”、“编程技巧”、“性能提示”、“可移植性提示”、“软件工程视点”和“测试与调试提示”; 每章以项目符号列表形式进行“小结”, 按字母顺序列出“术语”; 每章有“自测练习与答案”, 并有丰富的“练习”。

练习包括简单复习题、较长的编程问题到大项目。教师可以从第 3 章到第 21 章找到许多相关的问题作为项目练习。我们花了大量精力来设计这部分练习, 使本书对学生更有价值。

编写本书时, 我们使用了 Sun SPARC 工作站和 IBM PC 兼容机的各种 C++ 编译器 (Borland® C++ Builder™、Microsoft® Visual C++® 5.0 和 Metrowerks® CodeWarrior Professional®)。这些程序通常不需

要修改或只稍作修改即可在所有这些编译器中运行。我们的程序基于 Microsoft Visual C++ 进行开发 (少数例外已在文中注明)。

本书采用的 C++ 编程语言按照“信息系统国际标准草案工作稿——C++ 编程语言”(Working Paper for Draft Proposed International Standard for Information Systems——Programming Language C++) 中的定义。这个标准的批准和技术开发工作分别由认证标准委员会 X3、信息技术及其技术委员会 X3J16 (C++ 编程语言) 进行的。想获得详细信息请写信到下列地址:

X3 Secretariat
1250 Eye Street NW
Washington DC 20005

本书根据编程语言的美国家国家标准(American National Standard for Programming Languages)——C: ANSI/ISO 9899-1990 (Copyright 1990 by International Organization for Standardization) 定义的 C++ 编程语言编写, 该标准的副本可以直接从下列地址购买: The American National Standards Institute, 11 West 42nd Street, New York, NY 10036, 电话: (212)642-4900, 传真: (212) 398-0023, <http://www.ansi.org>。

真正的程序员应该认真阅读这些文献并经常参考。这些文献不是教程, 而是相当精确地定义了 C++ 和 C 语言, 满足编译器实现者和开发人员的需要。

我们认真对照了这些文献, 本书是初中级教程, 不可能介绍这些综合性文献中的每个特性。

教学目标

每章开头的“教学目标”介绍该章的学习目标, 告诉学生要学什么, 让学生学完该章后可以判断自己是否达到了这个目标, 帮助学生建立信心并使学习成果得到巩固。

节

每章分成若干个小节, 解决各个关键领域。我们喜欢分成许多小节。

248 个实例程序, 10660 行代码 (带程序输出)

C++ 的特性是在完整、可运行的 C++ 程序中提供的。每个程序后面是一个窗口, 包括程序运行的输出。这样使得学生可以确定程序如期运行。将输出与程序语句相联系是学习和巩固新概念的好方法。我们的程序使用各种 C++ 特性, 认真阅读本书就像亲临计算机前面输入并运行这些程序一样。

152 个演示 / 图形

书中包括大量演示插图。第 3 章介绍控制结构时提供了经过认真绘制的流程图 (注意: 我们不想把流程图作为程序的开发工具, 但可以通过流程图表示 C++ 控制结构的精确操作)。第 15 章“数据结构”用线图演示链表、队列、堆栈和二叉树的生成与维护。书中还有其他大量的程序演示。

621 个编程提示

我们用六个设计要素帮助学生注意程序开发、测试与调试、性能和移植性等重要方面。我们用“常见编程错误”、“编程技巧”、“性能提示”、“可移植性提示”、“软件工程视点”和“测试与调试提示”等形式列举了几百个提示。这些提示和实践总结了我们五十多年的编程和教学经验。最近, 一位数学系的学生告诉我们, 她觉得这些方法有点像数学教材中强调公理、定理和推论, 提供了建立良好的软件工程的基础。

115 个编程技巧

书中强调了编程技巧,要求学生注意掌握建立良好程序的习惯。向非编程人员讲课时,我们介绍编程的一个原则就是“清晰性”。我们告诉学生将在这些编程技巧中强调使程序更清晰、更易懂和更易于维护的技术。

201 个常见编程错误

学生学习语言时(特别是在第一门编程课程中)经常容易犯这样那样的错误。让学生注意这些“常见编程错误”有助于避免这些编程错误,还可以减少老师的答疑工作量。有些错误不是那么常见,但我们不想再加一类提示,因此也放在“常见编程错误”中。

88 个性能提示

根据我们的经验,让学生学会编写清晰易懂的程序是第一门编程课程最重要的目标。但学生可能想编写运行最快、使用内存最少、要求键击数最少或具有各种其他用途的程序。学生真正关心的是性能,希望为他们的程序快马加鞭。因此我们通过“性能提示”强调改进程序性能的方法。

35 个可移植性提示

软件开发是复杂而昂贵的活动。各种软件公司要经常开发对各种计算机和操作系统定制的版本,因此我们特别强调可移植性的问题,即产生的软件不做改变或稍做改变即可在各种不同的计算机系统中运行。许多人认为 C++ 是最适合开发可移植软件的语言,特别是 C++ 与 ANSI/ISO C 密切相关,ANSI/ISO C++ 草案标准又即将作为全球 C++ 标准而被批准。有人认为,只要用 C++ 编写实用程序,这个应用程序自然就是可移植的。其实不然,要实现可移植性,就要求认真细致的设计,需要避开许多陷阱。我们用“可移植性提示”帮助学生开发可移植的软件。

147 个软件工程视点

面向对象编程机制要求彻底重新考虑如何建立软件系统。C++ 是实现良好软件工程的有效语言。“软件工程视点”强调影响软件系统结构和构造(特别是大型系统)的技术、结构问题和设计问题。学生在这里学习的东西对高级课程和今后工作中建立大型、复杂的实际系统非常有用。

26 个测试和调试提示

“测试和调试提示”应该介绍测试程序缺陷和删除程序缺陷的建议,但书中大多数测试和调试提示实际上是关于防止缺陷进入程序的 C++ 功能和特性。

小结

每章最后有一些其他教学材料,包括丰富的项目符号列表形式的小结,帮助学生复习和巩固所学的关键概念。平均每章有 38 个小结子项。

术语

术语部分按字母顺序定义正文中遇到的术语,以便再次确认这些基本概念。平均每章有 66 个术语。

559 个自测练习与答案(包括各个部分)

大量“自测练习”和“自测练习答案”有助于自学,使学生可以建立信心,准备进行通常的练习。

862 个练习（包括各个部分，总共 1421 个练习）

每章最后有大量练习，包括简单概念和术语复习，编写各种 C++ 语句，编写小部分 C++ 函数和类，编写完整 C++ 函数、类与程序，以及编写大型项目。大量练习使教师可以根据学生的具体需要调整课程，每个学期指定不同课题。教师可以利用这些练习布置家庭作业、进行小测验和大型考试。

本书的快速浏览

本书分为几个部分。第一部分是第 1 章到第 5 章，详细介绍 C++ 过程式编程，包括数据类型、输入/输出、控制结构、函数、数组、指针和字符串。

第二部分是第 6 章到第 8 章，详细介绍数据抽象与类、对象和运算符重载。本部分也可以称为“对象编程”。

第三部分是第 9 章到第 10 章，介绍继承、虚函数和多态，是真正面向对象编程的根本技术。

第四部分是第 11 章到第 14 章，介绍 C++ 和面向流的输入/输出，包括流 I/O 与键盘、屏幕、文件和字符数组，并讨论顺序文件访问和直接访问（即随机访问）文件处理。

第五部分是第 12 章和第 13 章，介绍 C++ 中两个最新的补充，即模板和异常处理。模板也称为参数化类型，可以促进软件的复用性。异常处理功能帮助程序员开发更健壮、更具容错性、面向业务或任务的系统。

第六部分，即第 15 章，详细介绍链表、队列、堆栈、树等动态数据结构。本章加上第 20 章介绍的标准模板库可以生成丰富的数据结构，这些结构是 C++ 对传统计算机科学数据结构和算法课程的重要补充。

第七部分是第 16 章到第 18 章，介绍位、字符与字符串操作，预处理器和其他各种课题。

第八部分是第 19 章到第 21 章，介绍 C++ 最新改进和 ANSI/ISO C++ 草案标准中包括的 C++ 标准库。包括介绍 string 类、字符串流处理、标准模板库（STL）和 C++ 的其他最新改进。

本书最后的参考资料提供正文的附录，包括运算符优先级、ASCII 字符集、数值系统（二进制、十进制、八进制和十六进制数）和有关 C++ 的 Internet/World Wide Web 资源，并提供大量参考文献，帮助读者进一步钻研。下面详细介绍每章内容。

第 1 章“计算机与 C++ 编程简介”，介绍计算机是什么、如何工作、如何编程，介绍结构化编程的概念以及这些技术如何促进程序编写方式的变革。本章简要介绍了编程语言的发展历史，从机器语言、汇编语言到高级语言，讨论 C++ 编程语言的起源。本章还将介绍典型的 C++ 编程环境，简要介绍了如何编写 C++ 程序，并详细介绍了 C++ 中判断与算术操作的处理过程。学习本章之后，学生可以了解如何编写简单而完整的 C++ 程序。我们将介绍万维网与 Java 编程语言的出现及其所导致 Internet 的突飞猛进。我们讨论 namespace 和 using 语句，解释符合草案标准编译器的优点。我们继续在本版书的前几章中使用“旧式”头文件，而在后面出现大量新的 C++ 材料时则使用新式头文件。旧式编译器仍在广泛使用，几年之后才会完全退出历史舞台。读者如果想尽快进入面向对象的世界，可以看看每章“有关对象的思考”一节中关于对象技术的介绍。

第 2 章“控制结构”，介绍解决问题的算法（过程）概念，并解释在产生程序时如何有效利用控制结构使程序更易懂、更好调试、更易于维护，争取编译程序的一次性成功。本章介绍了顺序结构、选择结构（if、if/else 和 switch）和重复结构（while、do/while 和 for），详细讨论了重复，并比较了计数器控制循环和标记控制循环；介绍了产生结构合理的程序所需要的自上而下、逐步完善的方法，并介绍了常用的程序设计辅助伪代码。第 2 章使用的方法适用于任何编程语言中有效地使用控制结构，而不是只适用于 C++。本章帮助学生养成良好的编程习惯，以准备在本书后面处理更复杂

的编程任务。本章最后介绍逻辑运算符 `&&`、`||` 和 `!`。关键字表中增加了新的关键字，引入了新型的 `static_cast` 运算符，比从 C 语言继承的旧式 C++ 强制类型转换功能更安全。我们增加了 "Peter Minuit" 练习，让学生看到通过计算机程序计算复利的特点。我们对 C++ 源程序引入了更开放、更易读的“感观”效果。我们调整了模拟电梯的程序实例。根据读者反映，这个实例能够很好地把一些概念联系起来。我们对 `for` 循环中的循环计数器增加了新的范围规则。

第3章“函数”，介绍程序模块的设计与构造。C++ 中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、按值调用与按引用调用。第3章介绍的技术是建立结构合理的程序的关键，特别是对于系统程序员和应用程序程序员在实际应用中可能开发的大型程序和软件，这些技术更为重要。“分而治之，各个击破”的战略是解决复杂问题的有效方法，将问题分解为更简单的交互组件。学生对随机数和模拟器很感兴趣，他们很喜欢投骰子游戏，因为该游戏能很好地利用控制结构。本章介绍了递归技术，用一个表格总结了本书的几十个递归例子与练习。有些教材把递归放在后面讲，但我们认为这个课题应在讲课时由浅入深慢慢介绍。本章结束处的 60 个练习中包括几个经典的递归问题，如汉诺塔问题。本章介绍了“C++ 对 C 语言的改进”，包括内联函数、引用参数、默认参数、一元作用域运算符、函数重载和函数模板。这里已经修改了头文件表，并包括了许多新的头文件，读者将会在书中一一遇到。练习 3.54 在投骰子游戏程序中增加了赌注的功能。“有关对象的思考”一节介绍如何确定电梯模拟程序中类的属性（即数据）。

第4章“数组”，介绍由同一种类型的数据、一组数据和相关的数据所组成的数组。本章介绍了许多例子，包括单下标数组和双下标数组。众所周知，正确地构造数据和有效地使用控制结构一样，都是合理开发结构化程序的关键。本章的例子介绍各种常见数组操作、打印直方图、排序数据、将数据传递到函数，并介绍了调查数据分析（简单统计）。本章的一个特点是介绍了基本排序与查找技术并讨论了折半查找对线性查找的巨大改进。本章结束处的 38 个练习包括各种有趣而有一定难度的问题，如改进排序技术、设计机票订购系统、引入龟图的概念（在 LOGO 语言中很有名）和人工智能领域中非常著名的“骑士旅行”与“八皇后”的问题，引入启发式编程的概念。练习中有 8 个递归问题，包括选择排序、回文、线性查找、折半查找、八皇后、打印数组、逆向打印字符串和寻找数组中的最小值。本章仍然使用 C 语言式数组，这实际上是内存中指向数组内容的指针（第5章）。我们将把数组变为完全成熟的对象。第8章要用运算符重载技术建立 `Array` 类，生成比第4章的数组更健壮、更简单的 `Array` 类。第20章“标准模板库（STL）”中介绍 STL 中的 `vector` 类，在第20章使用迭代器及其算法时，可以用 `vector` 把数组变为完全成熟的对象。“有关对象的思考”一节介绍如何在电梯模拟程序中确定类的行为（即函数）。

第5章“指针与字符串”，介绍 C++ 语言中更强大也难于掌握的特性：指针。本章详细介绍指针运算符、按引用调用、指针表达式、指针算法、指针与数组的关系、指针数组和函数指针。C++ 中的指针、数组与字符串之间有非常密切的关系，因此我们介绍基本字符串操作概念，介绍一些最常用的字符串操作函数，如 `getline`（输入一行文本）、`strcpy` 和 `strncpy`（复制字符串）、`strcat` 和 `strncat`（连接两个字符串）、`strcmp` 和 `strncmp`（比较两个字符串）、`strtok`（将字符串标记化为各个组件）和 `strlen`（计算字符串长度）。本章结束处的 49 个练习包括模拟传统的龟兔赛跑、洗牌与发牌算法、递归快速排序和递归走迷宫。还有专门一节“建立自己的计算机”。本节介绍机器语言编程和一个设计与实现计算机模拟程序的题目，学生可以编写和运行机器语言程序。这个特性对于想了解计算机如何工作的读者特别有用。我们的学生很喜欢这个项目，因为经常能实现程序的重大改进，练习中就提出了许多改进。第15章另有一节介绍如何建立编译器，可以将编译器中产生的机器语言放到本章建立的机器语言模拟程序中执行，编译器与模拟程序之间的信息通过第14章介绍的顺序文件

进行通信。还有一个特殊小节,介绍有关文本分析、字处理、按不同格式打印日期、支票保护、写出支票金额大写、莫尔斯码和公制/英制换算的复杂字符串操作练习。学完第19章的string类后,学生要返回来复习这些字符串操作练习,许多人发现,指针问题是编程课程中最难的部分。在C和原始C++中,数组和字符串实际上是内存中数组和字符串内容的指针,即使函数名也是指针。认真学习本章将使读者深入了解复杂的指针问题。我们在本书稍后会把数组和字符串变为成熟的对象。第8章会介绍用运算符重载建立自定义的Array和String类。第19章介绍string标准库类和如何操作string对象。第20章介绍vector类。第5章有一些较难的练习,请试试“建立自己的计算机”一节。“有关对象的思考”一节介绍如何确定电梯模拟程序中类之间的关系。

第6章“类与数据抽象(一)”,开始介绍基于对象的编程思想。本章讨论如何正确进行数据抽象,即通过语言(C++)表达实现抽象数据类型(ADT)。近年来,数据抽象已成为讲解“算法课程”的主要课题。第6章、第7章和第8章详细介绍数据抽象。第6章将ADT实现为struct,实现ADT作为C++式的类,并介绍两种方法的比较以及使用C++式类的优点。还将介绍访问类成员,分开接口与实现方法,使用访问函数和工具函数,用构造函数初始化对象,用析构函数删除对象,通过默认的成员复制进行赋值和软件复用性。本章练习要开发复数、有理数、时间、日期、长方形、大整数和玩连城游戏的类。学生通常喜欢有关游戏的程序。“有关对象的思考”一节要求对电梯模拟程序中的每个类编写类的头文件和成员函数定义的源文件,其概述了运行电梯模拟程序的驱动程序操作,并提供一些有用的输出。对数字感兴趣的读者可以生成Complex类(对复数)、Rational类(对有理数)和HugeInteger类(对任意大的整数)。

第7章“类与数据抽象(二)”,继续介绍类和数据抽象。本章介绍声明与使用常量对象、常量成员函数,复合——建立以其他类对象为成员的类,友元函数与友元类对类的private和protected成员有特殊访问权限,this指针使对象可以知道自己的地址,动态内存分配,包含与操作类范围数据的static类成员,常见抽象数据类型的例子(数组、字符串与队列),容器类和迭代器。本章的练习要求学生开发一个存款账号类和保存整数集合的类。在讨论const对象时,我们简单介绍新的关键字mutable(第21章将会介绍),mutable用于修改const对象中“不可见”的实现方法。我们通过new和delete介绍动态内存分配。new失败时,它在旧式C++中返回0指针。第7章到第12章用这个旧式方法。第13章介绍新方法,在new失败时抛出异常。在static类讨论中增加一个基于视频游戏的例子。我们在教材中强调要把类的方法隐藏起来,不让客户看到。然后介绍类首部中的private数据,其显然要表示实现方法。我们另有一节介绍代理类,使类的客户看不到private数据。“有关对象的思考”一节要求在电梯模拟程序中加入动态内存分配与复合。学生非常喜欢生成Integerset类的练习,由此引入第8章介绍的“运算符重载”的概念。

第8章“运算符重载”,是C++课程中最著名的课题,学生很喜欢这方面的内容。他们发现运算符重载与第6章和第7章介绍的抽象数据类型非常匹配。运算符重载使程序员可以告诉编译器如何对新型对象使用现有运算符。C++已经知道这些运算符如何处理内部类型对象,如整数、浮点数和字符。但如果生成新的字符串类,加号是什么意思呢?许多程序员用加号表示字符串类的连接。第8章将介绍如何重载加号,使表达式中出现两个字符串对象时,编译器产生的函数调用“运算符函数”连接两个字符串。本章介绍运算符重载的基础、运算符重载的限制、重载类成员函数和非成员函数、重载一元与二元运算符以及进行类型转换。本章的特色是收集了大量实例,包括数组类、字符串类、日期类、大整数类和复数类(后两者在练习中有完整的源代码)。喜爱数学的学生可以在练习中生成多项式类。我们发现学生很喜欢该内容、这与大多数编程语言和教程中的做法有所不同。运算符重载是个复杂课题,但非常有意思。巧妙使用运算符重载可以使类变得更为实用。Array

和 String 类的介绍对使用 string 和 vector 标准库类的学生特别有用。利用第 6 章、第 7 章和第 8 章介绍的技术,可以建立一个 Date 类,如果过去使用过这个类,则现在很容易解决其中的“千年虫”问题。练习还鼓励学生用 Complex、Rational 和 HugeInteger 类增加运算符重载功能,像数学中一样方便地用运算符操作这些类的对象,而不是像第 7 章练习中使用函数调用。

第 9 章“继承”,介绍面向对象编程最基本的功能之一。继承是软件复用的一种形式,通过吸收现有类的功能再增加相应新功能而方便快捷地生成新类。本章介绍基类与派生类、protected 成员、public 继承、protected 继承、private 继承、直接基类、间接基类、基类与派生类中的构造函数和析构函数以及继承的软件工程。本章比较继承(“是”关系)和复合(“有”关系),引入“使用”和“知道”关系。本章的特色是几个重要实例,特别是用一个较长的实例实现点、圆、圆柱类的继承。本章最后介绍一个多重继承的实例,这是 C++ 的高级特性,使派生类可以继承多个基类的属性与行为。本章的练习要求学生比较通过继承与复合生成新类,扩展本章介绍的各种继承层次,对四边形、梯形、平行四边形、长方形和正方形编写继承层次,对二维形状和三维形状生成更加一般化的继承层次。我们对大学社区的成员修改继承层次,显示多种继承的例子。第 21 章继续介绍多重继承,讨论“菱形继承”问题和如何用 virtual 基类解决这个问题。

第 10 章“虚函数和多态性”,介绍面向对象编程的另一基本功能,即多态操作。多个类通过继承与同一基类相联系时,每个派生对象可以作为一个基类对象。这样就可以独立于派生类对象的具体类型,以更一般化的方法编写程序。新的对象类型可以用同一程序处理,从而使系统更可扩展。多态可以使程序消除复杂的 switch 逻辑,利用简单的直线逻辑。例如,视频游戏的屏幕管理器可以直接将绘图消息发送到要绘制的对象链表中每一个对象。每个对象知道如何绘制自己。新对象加进程序中时,不需要修改程序,只要新对象知道如何绘制自己就可以。这种编程方式常用于编写如今常见的图形用户界面(GUI)。本章介绍通过 virtual 实现多态操作的机制,区别抽象类(无法实例化对象的类)与具体类(可以实例化对象的类)。抽象类用于提供整个层次中类的可继承接口。本章的一个特色是两个重要的多态实例——工资单系统和第 9 章介绍的实现点、圆、圆柱类继承层次的另一种形式。本章练习要求学生讨论许多概念问题和方法,在形状层次中增加抽象类,开发基本图形软件包;修改本章的员工类,并要求在所有这些项目中使用虚函数和多态编程。本章的两个多态实例显示了相反的继承样式。第一个例子(工资单系统)是继承的“合理”用法。第二个是第 9 章介绍的实现点、圆、圆柱类继承层次的另一种形式,是专业人员所谓的“结构化继承”,不如第一个例子那么自然合理,但理论上是正确的。我们决定保留这两个例子,因为本书的这一版本中增加了多态、虚函数和动态关联技术的有关内容。我们曾经面向高级软件工程师举办了 C++ 专业研讨会,他们赞赏第一版中两个多态的例子,但觉得还缺了点什么。我们介绍了 C++ 中如何用多态编程,但读者还关心多态编程的操作开销。他们认为多态是个很好的特性,但显然是有代价的。因此我们的专业人士认为有必要更加深入地介绍 C++ 中如何实现多态,从而确定用多态编程时要付出的时间和内存代价。为此,我们建立了一个演示,显示 C++ 编译器支持多态编程样式时自动建立的 vtable(虚函数表)。我们在介绍点、圆、圆柱类层次的类中给出该表。与会人士认为这些表确实提供了对每个新项目确定是否适合用多态编程样式的信息。我们把这个例子放在 10.10 节, vtable 放在图 10.3。请认真观察该图,以便深入了解用继承和多态编程时对计算机的影响。

第 11 章“C++ 输入/输出流”,包含 C++ 中引入的新的面向对象式输入/输出的详细介绍。本章介绍 C++ 的各种 I/O 功能,包括流插入运算符、使用流读取运算符输入、类型安全 I/O(对 C 语言的巨大改进)、格式化 I/O、非格式化 I/O(对性能)、控制流的基数(十进制、十六进制或八进制)的流操纵算子、浮点数、控制域宽、用户自定义流操纵算子、流格式状态、流错误状态、用户自定

义类型对象的I/O以及连接输出流与输入流(保证要让用户输入响应之前出现提示)。本章有大量练习要求学生编写各种程序,测试文中的大多数I/O功能。

第12章“模板”,介绍C++语言演变中的一个最新补充。函数模板在第3章引入,第12章列举了更多函数模板的例子。类模板使程序员可以抓住抽象数据类型的本质(如堆栈、数组和队列),然后用少量补充代码生成特定类型的ADT版本(如int队列、float队列、字符串队列等等)。为此,模板类通常称为类型参数。本章介绍使用类型参数和无类型参数,考虑模板与其他C++概念之间的交互,如继承、友元和static成员。练习要求学生编写各种函数模板和类模板,并在完整程序中采用这些模板。这一版的第12章没有任何补充,但我们在第20章介绍了标准模板库(STL)容器、迭代器和算法,大大改进了模板处理。

第13章“异常处理”,介绍C++语言中一个更新的改进。异常处理使程序员可以编写更健壮、更具容错性、更适合业务关键和任务关键情况的程序。本章介绍何时需要异常处理,介绍异常处理的基础,try块、throw语句和catch块,介绍何时、如何重新抛出异常,介绍如何编写带有异常处理和不希望异常处理的进程,介绍异常与构造函数、析构函数、继承之间的重要关系。本章提供了43个练习,通过程序演示C++异常处理功能的强大和丰富多彩。我们在图13.2的例子中增加了重新抛出异常的“有生命力的代码”,图13.3增加堆栈解退的“有生命力的代码”。我们在图13.4和图13.5中增加两个例子,演示内存不足时new失败的两种情况。在C++草案标准之前,new失败返回0,就像C语言中malloc失败返回NULL指针值(如图13.4)。图13.5显示了新的样式,new失败返回bad_alloc异常。我们用图13.6的例子演示如何用set_new_handler指定处理内存不足情况的自定义函数。我们增加13.15节,讨论新的auto_ptr类模板,保证正确地释放动态分配的内存,以防止内存泄漏。我们还在13.16节讨论新标准库的异常层次。

第14章“文件处理”,介绍用顺序访问和随机访问方法处理文本文件的技术。本章首先介绍从位、字节、域、记录到文件的数据层次;然后显示C++的简单文件和字符流视图。顺序访问文件用三个程序讨论,显示如何打开和关闭文件、如何在文件中顺序存储数据和如何从文件中顺序读取数据。随机访问文件用4个程序讨论,显示如何顺序生成随机访问文件,如何顺序读取和写入随机访问文件,第4个随机访问程序组合了许多顺序和随机文件的访问方法,建立一个完整的事务处理程序。专业研讨会的学员告诉我们,学习文件处理的内容之后,他们可以建立相当大的文件处理程序并立即在公司使用。练习要求学生实现各种程序,建立和处理顺序访问文件和随机访问文件。第14章删除了字符串流处理的有关内容,我们把这个内容改写之后放在第19章末尾。

第15章“数据结构”,介绍生成和操作动态数据结构的技术。本章首先介绍自引用类和动态内存分配,然后介绍如何生成与维护各种动态数据结构,包括链表、队列(或等待队列)、堆栈与树。对每种数据类型,我们介绍完整的、可工作的程序,并显示示例输出,实际地帮助学生掌握指针。本章的大量例子使用间接地址和双间接地址,这是特别难掌握的概念。使用这种指针时的一个问题是学生很难将数据结构及其节点链接方式直观化。因此我们用图形显示链接及其生成顺序。有关二叉树的例子特别适合演示指针和动态数据结构。这个例子生成二叉树,强制消除重复元素,并引入递归的前序遍历、中序遍历和后序遍历的树遍历方法。学生学习和实现这个例子之后会很有成就感,他们特别欣赏中序遍历方法按排序顺序打印节点值。本章包括大量例子,特别推出“建立自己的编译器”练习。这个练习让学生开发中缀/后缀转换程序和后缀表达式求值程序,然后修改后缀表达式求值算法,产生机器语言代码。编译器将这个代码放到一个文件中(用第1章介绍的方法)。然后学生在第5章练习中建立的机器语言模拟程序上运行编译器产生的机器语言。本章的35个练习包括一个使用队列的超级市场模拟程序,递归查找链表,递归打印反序链表,删除二叉树的节点、