

# C++ 程序设计

(第三版)

## C++ PROGRAM DESIGN

(Third Edition)

[美] James P. Cohoon 著  
Jack W. Davidson

刘瑞挺 韩毅刚 等译  
盛素英 刘海嘉



Education

<http://www.mheducation.com>



电子工业出版社

Publishing House of Electronics Industry

<http://www.phei.com.cn>

# C++程序设计

(第三版)

C++ PROGRAM DESIGN

(Third Edition)

[美] James P.Cohoon 著  
Jack W.Davidson

刘瑞挺 韩毅刚 等译  
盛素英 刘海嘉

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



麦格劳-希尔教育出版集团

## 内 容 简 介

本书是美国大学新生学习计算机入门课程 (CS-1) 的教材, 既适合计算机专业, 又特别适合非计算机系的其他各个专业。选择 C++ 的理由有二: 一是当今大多数应用软件都使用 C++ 编写而成, 二是 C++ 最能体现面向对象程序设计的思想。这样作者就把编程语言与编程思想融为一体, 在一步步学习 C++ 编程的同时, 读者便一层层掌握了面向对象的设计原理。本书共有 15 章, 分别介绍了计算与面向对象设计方法学、C++ 基础、修改对象、控制结构、函数基础、程序员定义的函数、类结构与面向对象的设计、抽象数据类型的实现、列表、EzWindows API 的详细分析、指针和动态内存分配、测试与排错、继承、模板与多态性和软件项目等。本书内容翔实、叙述清晰、注重实践、习题丰富, 适合作为大专院校各个专业的计算机入门教材, 也适合各领域想在计算机方面进修提高的人员自学使用。

COHOON & DAVIDSON: C++ PROGRAM DESIGN, Third Edition.

ISBN 0-07-241163-5

Copyright 2001 by the McGraw-Hill Companies, Inc.

Original language published by The McGraw-Hill Companies, Inc. All Rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and Publishing House of Electronics Industry.

本书中文简体字翻译版由电子工业出版社和美国麦格劳-希尔教育 (亚洲) 出版公司合作出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。版权所有, 翻印必究

本书封面贴有 McGraw-Hill 公司激光防伪标签, 无标签者不得销售。

# 译者序

我国程序设计的教学，从非计算机专业的计算机课程角度看，已经走过了 30 多个春秋。自 20 世纪 70 年代始，非计算机专业的人士开始学习计算机程序设计。当时学习者还不多，主要是学汇编语言以及高级语言 ALGOL-60。由于上机条件比较差，纸上谈兵者或纸上穿孔者均不乏其人。

进入 20 世纪 80 年代后，中央电视台播出了谭浩强教授的《BASIC 语言》课，掀起了学习计算机的热潮，使许多不懂计算机的知识分子扫了盲。后来，随着微机的大量涌入，上机条件有所改善。通过实践，人们开始重视程序设计方法的问题，例如认识到非结构化的、无序 Goto 带来的麻烦与危害，澄清了结构化程序设计的必要性。与此同时，软件工程也理所当然地进入了非计算机专业的课堂。

从 20 世纪 90 年代开始，简单方便的 Windows 界面和 Office 套件吸引了绝大多数初学者的注意力。正如当年人们曾以为 BASIC 就是计算机那样，Windows 也成了计算机的代名词，而且 Word 成为初学者入门的第一课。这个趋势是无可非议的，然而这一倾向却掩盖了另一个倾向，那就是许多人在程序设计上犹豫、徘徊，似乎非计算机专业的人大可不必学习程序设计了。

新世纪为中国展现了全面发展的大好前景，中国与世界接轨的伟业正在铿锵有力地进行。在这种形势下，我们有理由重新审视那个曾经争论不休的问题：广大的非计算机专业人员当真不需要学习程序设计技术吗？显然，他们需要学习程序设计。否则，他们就不可能在自己的领域创造性地应用计算机技术。先进国家的经验也证实了这一点。为此，我们满怀信心地翻译了这本书，希望它能使计算机教学更顺利地与先进国家接轨。

这是一本令人称赞的好书，原书的全名是《C++程序设计，编程与面向对象设计导论》（C++ Program Design, An Introduction to Programming and Object-Oriented Design）。它是美国弗吉尼亚大学的考恩（James P. Cohoon）教授和戴卫森（Jack W. Davidson）教授为计算机科学教育而开发的第一门课程（CS1），该教材对全校各个专业都十分合适。但它同样可以用于其他课程。例如纽约城市大学的麦克拉肯（Daniel D. McCracken）教授就使用本书作为软件设计实验室课程的教材。

那么，这本书有什么特点呢？我认为它有“四个面向”的特点：即**面向非计算机专业、面向对象程序设计、面向软件实践基础、面向读者自学提高**。

第一，这是一本面向非计算机专业的计算机入门教科书，虽然它也可以作为计算机专业的教材。我们知道，在整个专业教育中，有 95% 的人是从从事非计算机专业的，而这些专业都毫无例外地存在着应用计算机的广阔天地。作者认为，在今天的信息时代，无论哪个专业的人都应该学会一点程序设计，而作者选择的编程语言正是 C++。这种选择的依据是：在各种领域的应用软件设计中，C++ 已经成为不可或缺的工具；在各行各业的招聘面谈中，能否熟练掌握 C++ 语言已经成为择优录取的重要条件。过去，人们以为只有计算机专家才会 C++，只有计算机专业才需要学习 C++，而非计算机专业的人根本用不着学 C++，这种观点已经与现实世界大相径庭。这是作者在设计这门课程时，进行了大量社会调查后得出的结论。

第二，这本书充分强调面向对象的程序设计，而根本不涉及面向过程的程序设计。正如 C++ 不是计算机专业的专利那样，面向对象程序设计也不是计算机专业的特权。许多人误以为面向对象的概念十分抽象，什么封装啦、继承啦，太难懂了。其实，这完全是一个误导。我们知道，面向对象这个革命性的计算概念是今天大多数高级语言的基石，C++ 的基石是面向对象，Java 的基石也是面向对象。面向对象编程技术的缔造者、电脑未来学家艾兰·凯 (Alan Kay) 在 Xerox 的 PARC 研究中心时，创造了一种编程语言 Smalltalk，它是面向对象的起源。凯喜欢把它比喻为一个生物学系统，例如人体系统。人体是由数以万亿计的细胞组成的，每个细胞都要执行自己的特定任务。每个细胞并不知道其他细胞内部发生的事情，但它们能互相沟通，并且协作完成更加复杂的任务。每个细胞用细胞膜封装起来，彼此却能通信联系，并且通过遗传来继承前辈的功能。因此，面向对象与面向过程是两种完全对立的方法。过去的一些书，总是先大讲特讲面向过程，而后再少讲略讲面向对象，以为这是全面系统的教学方法。殊不知，这种折衷恰好冲淡了面向对象的精髓。而本书则开门见山地树立对象概念，反反覆覆地巩固对象概念，使读者能真正学会这种优秀的编程思想与方法。

第三，本书非常重视实践环节。它提供的 EzWindows 看来是一个大赢家，对学生有很大的启发性，使他们看到响应鼠标点击就能实现作图，这正是他们编程的方式。本书还为最后的项目提供了良好的跳板，通常在课程最后花 3 周甚至 4 周的时间，让两个学生一组工作。在 CD-ROM 光盘上，存有学生所需的全部代码和 EzWindows 库函数，这节省了学习的时间。本书的每个细节，从组织到编程技巧，都为今后的软件工程课奠定了牢固的基础。

第四，本书的写作风格特别适合于自学，更不必说当作听课的教材了。它细腻而不失于简练，翔实而不落于烦琐，连贯而不陷于空泛，使人读来爱不释手。第 2 版就有大量很好的练习，从相当简单的评论是非，到大多数学生要花若干小时的编程项目。第 3 版又增加了许多自我检测题，使读者能验证自己掌握的程度。作者在前言里或者个人谈话中，都没有要求讲授或自学全书的内容，在弗吉尼亚大学的第一门课程即是如此。教师可以根据自己的特殊需要，自由选择想要强化的内容。作者在前言里给出他们在弗吉尼亚上第一门课的安排，本教材作为 CS1 可以有許多不同的处理方式，也可以有许多种自学的方法。的确，本书能以完全不同的方式使用。显然，人们喜欢这本书。经验证明，它使教学获得乐趣，而且听不到学生的抱怨，这是十分罕见的现象。

参加本书第 2 版翻译的有刘瑞挺、宋杏珍 (前言、序言、第 1 至第 3 章)，韩毅刚 (第 4 至第 8 章)，盛素英 (第 9 至第 12 章)，刘海嘉、刘璟 (第 13 至第 15 章、附录 A 至 F)。刘瑞挺教授、刘璟教授曾对第 2 版整个译稿作了审校。参加第 3 版翻译修订的有刘瑞挺 (前言、译者的话、第 1 至第 3 章)，韩毅刚 (第 4 至第 12 章，第 12 章测试与排错是第 3 版新加的)，赵永欣 (第 13 至第 15 章)，肖文俭 (附录 A 至 F)。刘瑞挺教授对第 3 版整个译稿作了审校与润色。我们感谢生物学家刘瑞林教授、凌元洁教授，帮助翻译了书中有关恐龙分类的专业名词。我们还要特别感谢电子工业出版社，如果没有他们的督促、鼓励、建议以及辛勤的编辑加工，这本书就不可能如此迅速地与广大读者见面。由于译者才疏学浅，不当与谬误之处在所难免，敬请读者不吝指教，以期再版时更正。

刘瑞挺 2001 年初冬 北京

# 前 言

## 引言

计算机是我们生活中不可或缺的设备，它控制着复杂的系统，诸如金融网络、公共交通、电话系统和发电站等等。千百万人通过因特网获取信息、购物消费、休闲娱乐、进行通信及处理商务。由于计算机已成为现代生活真正含义上的组成部分，所以我们确信：每个人都应该具有为计算机编程的基础知识。

这本教材是使用 C++ 开发软件与编程的基础，而 C++ 是由 AT&T 贝尔实验室的 Bjarne Stroustrup 建立的一种流行的高级编程语言。我们选择 C++ 讲授编程是因为它支持面向对象的方法开发软件。面向对象方法的优点是：我们能用它开发复杂的软件系统，而它所用的许多技术是与建造复杂的物理系统，诸如制造汽车、飞机、或者盖大楼时曾经用过的技术类似。本书的目标是把它作为大学生学习编程的第一门课程，而且设计成适合所有专业的学生。我们假定他们预先没有学过编程技能，而所用的数学与理科知识也仅限于一年级大学生的水平。

本书的一些重要特点如下：

· **广泛涵盖 C++ 标准。** 我们的初衷是覆盖 C++ 的整个范围。但这样冗长的叙述是大学新生难以承受的。例如，语言定义描述了 150 多个标准类与标准库。看来，与其使它成为百科全书，还不如提供导论课所需深度的资料。我们介绍了这类资料，而把指针放到后面处理。我们还提供了 C++ 语言的重要修改与增强，如布尔 (bool) 类型、标准模板库、名字空间、例外等。我们覆盖的广度为教师提供了灵活性。例如，教师可以不选择继承 (inheritance)，而选择模板。对学生来说，这种覆盖范围也给他们进一步学习语言带来好处，而且也使本书具有参考书的价值。

· **尽早引入类的概念。** 第 1 章我们简要介绍了面向对象的范式，提供了激发学生学习欲望的材料。我们相信，大学生在尚未领会设计那些灵活而有用的对象将遇到的种种困难之前，他们首先应该成为对象的客户端用户。所有程序设计高手都是从用户开始的。接下去的几章，我们介绍并使用了一些标准的流类对象，如 cout 和 cin，使用了标准模板库的 string 对象，以及为本教材开发的图形库中那些数量有限的对象。这些经验有助于增强对封装、软件重用，以及面向对象编程范式的理解。在巩固了对象的这些概念后，我们提供了 50 个类，以及在最后 8 章用到的抽象数据类型。

· **为了初学者能开发趣味程序，我们专门设计了图形应用编程界面 (API)。** 我们提供了简易的面向对象图形库 EzWindows，便于显示简单的几何、位图以及文本对象。我们为流行的 Windows、UNIX 和 Macintosh 编译程序补充了 API 的实现，为大学生使用 API 提供了许多重要的经验。首先，大学生是软件库的客户端用户。如前所述，使用设计良好的对象，有助于编程初学者有一个面向对象设计的良好开端。他们有了作为用户的经验，这可以成为未来设计师的基础；其次，使用 API 可以引导学生进入开发程序的现实世界，通过专门应用库进行开发实践；第三，使用 EzWindows 完成图形输入输出，给学生阐明了基于事件的编程、在实际应用中占支配地位的输入输出模式，以及能使他们开发有趣的、令人兴奋的可视化程

序。这个经验激励着学生的学习欲望，提供的分立对象集可以帮助学生理解面向对象方法的范式。EzWindows 相当简单，使得第一次上编程课就能实现图形化。使用 EzWindows 的例子在本教材中通篇皆有。但是，我们表达的流行方式是希望教师喜欢 ANSI 的材料。对于想要更高级图形能力的教师和大学生，他们可以参考附录 E，那里有完整的 EzWindows 材料。如果需要示范课程与辅助资料，请访问我们的 Web 站点 <http://www.mhhe.com/c++programdesign> 和 <http://www.cplusplusprogramdesign.com>。

- **通过软件项目和案例研究引入软件设计的概念。**本书除包括引进 C++ 与面向对象设计概念的许多小例子外，每章都详细地考察了一个或者更多的实际问题。与此相应，我们还考虑了面向对象的分析与设计、算法开发，以及实现这些设计的编码。此外，有两章（第 10 章和第 15 章）专门讨论了使用 EzWindows API 进行软件项目开发的原理。这些章节是通向软件重用以及开发适合个人与小组工作的软件项目的必由之路。

- **与教材正文清楚区分的编程风格和技巧资料则表示在带底色的框中。**为了增加对 C++ 和面向对象编程的理解，我们还就怎样成为具有更多更好知识的程序员和设计师给出建议。例如，避免一般编程错误的重要技巧、编写易读的代码、理解纳入标准的新方向，以及涉及性能与软件工程的技巧。

- **整合了标准模板库的使用。**C++ 语言有一个重要部分，这就是标准模板库，或者更常把它简称为 STL。STL 将改变程序员开发软件的习惯。这个库提供了丰富的、为表示串与表的容器类，或称集装箱类（container），还有为完成重要编程任务如排序、查找，以及能遍历表的各种算法。读者首先会在第 3 章介绍字符串类时遇到 STL，接下来的几章给出补充的特征。特别是第 9 章利用 STL 向量容器类彻底地探讨了表的表示，第 11 章和第 14 章察看了 STL 容器类如何实现，以及使用通用算法如何解决一系列普通的编程任务。

### 第三版的要点

本书第三版接受了来自教师与学生的许多建议。以下是值得注意的改进与补充：

- **自我检测题。**在每一章的适当位置，安排了自我检测题，学生用它可以检测其对教材掌握的情况。自我检测题补充了每章末尾的练习题。自检题的答案，请访问我们的 Web 站点：<http://www.mhhe.com/c++programdesign> 和 <http://www.cplusplusprogramdesign.com>。自检题既有答案简短的问题，也有编写程序的练习。

- **较早出现类的内容。**类的内容出现比较早（第 7 章）。这个变化反映了增长的一致性，早期提供类的内容是可能的，也是用 C++ 讲授面向对象编程的正确途径。

- **输出测试和排错的内容。**程序员的一个重要技能是如何对他编写的程序进行测试和排错，本书过去的版本没有包括这个重要的论题。第 12 章提供了对测试和排错的概述，它讨论了各种测试技术，例如单元测试、综合测试，以及编码检验。排错部分着重教给学生如何用科学的方法找到错误。这一章还讨论了初学程序员遇到的常见错误，以及如何识别它们。

- **改善了对问题的解释。**根据我们自己的分析和读者的反馈，对整个教材进行了重新考查，增加了补充的例子，重画了清晰的插图，并相应地校订与扩充了文字解释。

### 内 容

20 世纪 90 年代初期，在（美国）国家科学基金（NSF）的支持下，弗吉尼亚大学计算机

科学系着手开发新的计算机科学的课程设置。我们谨慎地考查了本校以及许多兄弟院校已有的课程设置。

我们发现，这些课程设置强调的是：

- 采用的编程语言只为给学生上课，而在实践中极少使用。
- 建立的小程序，最多只有数百行组成。
- 开发基于文本的程序，每个任务都孤立地“从零开始”。
- 开发环境缺乏现代工具。
- 非正式的开发，其理念是如果程序能“工作”，它就可以接受。

把这种情况与现实世界比较，我们看到值得深思的差距。实际需要的计算机专业知识是：

- 使用编程语言设计并开发大型应用程序，它们的源代码通常有数千行甚至上百万行。
- 不仅参与开发，更要经常参与对这些系统的修改与维护。
- 在团队中工作，而不是作为一个程序员单干。
- 按照指令性的规格说明进行系统开发。
- 建立的系统要使用图形用户界面进行输入与输出。
- 使用已有的库与工具去建构系统。

为了使大学生对真实世界的编程作好准备，我们编写了本书的第一版。第三版反映了我们收到的、来自教师与学生两方面的反馈意见，以及我们自己使用本书的经验。我们用它教过大型的编程绪论课，以及关于编程与软件工程的后续课。

## 编 程

如果对什么是程序 and 如何编写程序没有很好的了解，那么对计算机科学中的大多数概念与问题就难以领会。不幸的是，编写程序不好学。编好程序，就像写好文章一样，需要多年的苦练。事实上，讲授编程与讲授写作，在某些方面是非常类似的。

大学生学习写作要通过阅读优秀散文范例，并且要通过写作、写作、再写作。在这个过程中，他们学习了组织思路的重要技巧，于是就能进行有效的表达。随着学生技能的发展，他们由写作与编辑一个段落或若干个段落，进而能创作大块的文章，如随笔、短篇小说和报告等。

我们讲授编程的方法类似于讲授写作，但有一点非常重要的补充。贯穿本书，我们叙述与讨论了许多编程的例子，好的和坏的例子都有。编程练习给学生提供了组织与编写代码的实践机会。另外，我们还提供了例题，方便学习修改已有代码的实际技能。这里使用的代码是专门为学生进行修改而设计的。我们发现这种机制是有效的，因为它强迫学生阅读并理解提供的代码。在正文中，CD-ROM/World Wide Web 图标表明，该代码在本书附带的 CD-ROM 光盘（需另购）里提供，或者通过访问我们的 Web 站点可以得到。

## 为什么用 C++?

在我们着手开发新的课程设置时，首要议题之一就是选择编程语言。像许多计算机系那样，我们使用过 Pascal 语言。虽然必须换掉 Pascal 的决定是毫无异议的，但选择什么语言来代替却有相当热烈的争论。我们考虑过的语言有 C、C++、Modula-3、Scheme 和 Smalltalk 等等。决定的因素是，我们要用的语言应该是我们自己的专业经常要使用的。这个决定使选择

集中到 C 或者 C++ 上。尽管决定选择 C++ 并非没有异议，但我们确信面向对象的范式将是未来编程中占优势地位的范式。

事后证明，我们的选择是正确的。C++ 持续不断地流行起来，而且许多公司用它作为开发语言。的确，我们的许多研究生说，他们在求职面试时，经常被问及的一个问题就是会不会 C++。我们相信，入门性编程语言的选择将继续向 C++ 转移。我们还由衷地为教过的学生取得的效果而感到惊喜。他们在后续的课程中，已经完成了我们的软件开发系列，比起按旧的课程设置培养的学生，他们处理的问题更大、更难。另外，我们还看到其他学科向 C++ 的实质性转移。例如，我们学校商学院的所有学生现在都学 C++，工学院的学生过去需要 Fortran，而现在也需要 C++。

你们也许有人考虑是不是用 Java。Java 的确是饶有兴趣的语言。但是，作为重要的软件开发工具还不成熟，该语言及其库还在进行重大的修改。在我们的研究领域内，专业人士的明智看法是 Java 可能适合开发图形用户界面 (GUI)，而 C++ 属于开发应用程序的语言。如果我们要求的目标是最能把学生培养成计算机专业人才，那么 C++ 就是正确的教育工具。

## 早些引入对象

我们在过去八年讲授 C++ 的经验表明，面向对象的范式可以成功地介绍给初学编程者。在我们原先提供的课程中，只是在课程接近结束时才介绍对象的概念，并肤浅地覆盖了对象、类、重载、继承等。基本上，我们还是讲 C，只不过使用了 C++ 的语法和输入输出机制而已。这个方法失败了。在课程中它介绍的新概念太晚了。学生们不能综合这些材料。我们修改了课程，比较早地引入对象，发现这样做的效果要好得多。学生们现在有时间吸收这些材料，因为他们不是在课程结尾，而是贯穿全课程都在使用并加强它。在本教材中，反映了尽早接近对象的想法。在第 2 章，学生们开始使用标准对象。从第 3 章到第 7 章，引导学生通过 EzWindows API 使用图形对象。在使用这些对象巩固之后，第 7 章引进了类与对象设计，这在逻辑上跟进了第 4 章到第 6 章引进的控制结构、函数，以及库。我们坚定地相信，在导论式的教科书中，如此安排材料的顺序是恰当的。学生们当然喜欢它！他们通过第二次和第三次课程设计，就能做出带图形功能的有用软件。

## 软件项目

显然，我们过去所教的东西并非完全局限于真实世界所发生的事情上。为了培养未来的计算机科学家有能力支持大型软件系统工程、重建已有的系统，以及使创新的技术得到应用（如软件的重用），我们系认为，在第一门课程中必须引进这样的项目材料。在我们的软件项目章节中（第 10 章和第 15 章）推出了这样的介绍。这些章给学生们提供了许多重要的经验。首先，这两个项目使用了我们的 EzWindows API。使用 API 去做基于事件的编程和图形输入输出，给学生们展示了在真实世界的应用中使用的典型编程模型，这使学生能开发更加有趣、更加惊喜的程序。如果想要的话，这些软件项目可以方便多至四人一道工作。这个实践再次反映了真实世界的情况，即极少有单个程序员去开发应用程序。软件项目还演示了软件的维护。软件章节的末尾有许多经验，要求学生项目的程序进行重要的修改或不重要的扩充。

## 各章摘要

### · 第 1 章：计算与面向对象设计方法学

基本计算术语，机器组织，软件，软件开发，软件工程，以及面向对象设计与编程。

### · 第 2 章：C++基础

程序组织，main()函数，包括语句、注释和定义，编写可读代码、交互输入输出，基本类型，字面，常数，说明，表达式，转换，以及优先级。

### · 第 3 章：修改对象

赋值语句与转换，提取，const 对象，增量与减量，插入与提取，字符串类，标准模板库，图形对象，以及 EzWindows API。

### · 第 4 章：控制结构

逻辑值与运算符，真值表，布尔，关系运算符，一般优先级，短路求值，if 语句，if-else 语句，排序，开关语句，enum 语句，while 语句，for 语句，不变量，do 语句，文本处理，以及科学可视化。

### · 第 5 章：函数使用基础与库

函数，值参，形参，实参，调用，控制流，活动记录，伪随机数，原型，预处理器，包含指令，头文件，条件编译，软件重用，使用库，标准流，操作符，文件流，文件处理，iostream，iomanip，fstream，math，ctype，string，stdlib，以及 assert 库。

### · 第 6 章：程序员定义的函数

函数定义，参数，调用，控制流，return 语句，作用域，局部对象，全局对象，引用参数，常量参数，默认参数，参数分配，函数重载，初始化，名字重用，自顶向下设计，递归，存储流，实用程序函数，标准模板库，二次多项式的积分，以及金融可视化。

### · 第 7 章：类结构与面向对象的设计

程序员定义的数据类型，类结构，信息隐蔽，封装，面向对象的分析与设计，访问说明，数据成员，成员函数，构造函数，万花筒程序，面向对象的工厂自动化模拟程序与训练程序。

### · 第 8 章：抽象数据类型的实现

数据抽象，面向对象设计，默认与拷贝构造函数，检查器，转变器，简化器，辅助函数，成员赋值，const 成员函数，算术运算符重载，引用返回，插入与提取重载，伪随机数发生器，对有理数与伪随机数的 ADT，以及红黄绿游戏。

### · 第 9 章：列表

一维数组，下标，参数传递，初始化，字符串，多维表，表，矩阵，标准模板库，容器类，适配器类，向量类，向量成员函数，排序，InsertionSort，QuickSort，二叉搜索，二维搜索，表的表示，初始化列表，迭代，以及对机器人迷宫问题的 ADT。

### · 第 10 章：EzWindows API：详细考查

应用程序员界面 API，图形用户界面，基于事件的编程，窗口坐标系统，回调，鼠标与定时器事件，EzWindows API 机制，单窗口 ADT，位图，文本标号，以及 Simon Says 游戏。

### · 第 11 章：指针与动态内存分配

左值 lvalues，右值 rvalues，指针类型，寻址，间接寻址，作为参数的指针，指向指针的指针，常量指针，数组的等价与指针的记号，字符串处理，命令行参数，指向函数的指针，

动态对象，自由存储，new 与 delete 运算符，悬空指针，存储器泄漏，析构函数，成员赋值，this 指针，以及整数列表的 ADT。

· 第 12 章：测试与排错

黑盒测试，白盒测试，检查，单元测试，集成测试，系统测试，语句覆盖，等值分区，回归测试，边界条件，代码审查，测试探针，以及路径覆盖。

· 第 13 章：继承

面向对象设计，重用，基类，派生类，单继承，is-a 关系，has-a 关系，uses-a 关系，图形的层次，控制继承，保护成员，多继承，对长方形、圆形、椭圆形和三角形的 ADT，以及面向对象的万花筒程序。

· 第 14 章：模板与多态性

通用行为与类型，函数模板，类模板，容器类，顺序表，链表，迭代类，友元，多态性，虚函数，纯虚函数，抽象基类，虚派生类，虚多继承，表 ADT，随机访问表，顺序表，迭代表，单链表，以及双链表。

· 第 15 章：软件项目——追捕虫子！

封装，继承，虚函数，面向对象设计，追捕虫子游戏，以及各种类型虫子与游戏控制器的 ADT。

· 附录

ASCII 字符集，一般优先级表，iostream、stdlib、time、string 和算法库，向量与其他容器类，串类，名字作用域，using 语句，异常，友元，EzWindows API，项目，以及文件制作。

## 本书的使用

这本教材比单一课程包含的资料更多。超出的内容都是深思熟虑的，以便教师对于编程和软件开发的课题进行选择。本书还对教课的灵活性进行了设计。例如，如果教师希望在课程中更早地介绍类的内容，他/她就可以在类和我们开发的有理数 ADT 后介绍迭代。如果教师希望在数组之后再介绍类，那么第 9 章的 9.1 至 9.5 节和 9.12 节就可以提到第 7 章和第 8 章的前面。还有，第 13 章对继承的讨论可以提到第 11 章关于指针和动态对象的前面。对计算机科学内容希望广度优先的教师可以省略软件项目的章节，并用诸如计算机的社会与道德问题或者基本的形式逻辑内容取而代之。第 12 章的测试材料（12.1 节）可以安排在关于类的材料（第 7 章）之后的任何时间进行。排错的内容（12.2 节）依赖于数组的例子，因此应该安排在第 9 章和第 11 章之后。

我们的课程内容安排如下：

周	内 容	阅 读
1	计算与面向对象设计	第 1 章
2	编程基础	第 2 章
3	对象控制	第 3 章
4	条件语句	第 4 章（4.1 至 4.6 节）
5	迭代语句	第 4 章（4.7 至 4.12 节）
6	函数与重用	第 5 章，第 6 章

7	参数传递	第 6 章
8	OO 分析与设计	第 7 章
9-10	ADT	第 8 章 (8.1-8.6 节)
11	数组	第 9 章 (9.1-9.4 及 9.12 节)
12	向量	第 9 章 (9.5-9.10 节)
13	项目——西蒙说, OOA/OOD	第 10 章
14	继承	第 13 章


按照每位同行的兴趣, 这 14 周的材料可以变更。在我们学校的导论课程中, 每个学期都要花 1 周的时间对问题作详细说明。一般说来, 考试取决于最后的项目。

### 辅助材料

在附加的 CD-ROM 中, 包括了我们开发的许多程序与列表的源代码和辅助文件。例如, 有一整套清晰的幻灯片 (约 300 张)。我们讲授的课程还有密切结合实验的内容, 每周一次以满足对当前课程的巩固。对于这些实验室工作, 我们给学生开发了实验室手册, 这些材料可以从出版社得到。对于更详细的信息, 请访问我们的 Web 站点 <http://www.mhhe.com/c++programdesign>。特别值得说明的是我们维护了一个常见问题表 (FAQ), 并链接到许多有用的 C++ 及教育网站。其他教育辅助材料还可以从我们的课程 Web 站点得到: <http://www.cs.virginia.edu/cs101>。

### 符号说明

本书通篇均在页面边缘使用了下述图标。

 WWW 和 CD-ROM 图标, 它与某些程序和代码列表有关。该图标表明, 这些代码可以从本书所附光盘或者访问我们的 Web 站点得到: <http://www.mhhe.com/c++programdesign>。当该图标与一个带标号的程序有关时, 表明该程序由单个文件组成。如果该图标与一个带标号的列表有关时, 表明可用的是一个库文件或者是在多文件程序中的一个文件。



惊叹号图标是对编程的警告, 通常这表明避免一般编程错误的技巧。



绕路图标表明是一组自检练习, 练习的答案可以在我们的 Web 站点得到: [www.mhhe.com/c++programdesign](http://www.mhhe.com/c++programdesign) 或者 [www.mhhe.com/cplusplusprogramdesign](http://www.mhhe.com/cplusplusprogramdesign)。自检练习既有简短答案的问题, 也有编写程序的练习。



太阳镜图标表明是与编程风格有关的材料。目前, 有大量的习惯用法。一般说来, 本书采用的编码风格反映了占优势的用法。(当然, 我们的改动是最佳的!)



书本图标是与 C++ 编程语言有关的材料。该图标有两个典型用途，一是说明为 C++ 的高级论题，另一是说明对当前语言的扩展，这些扩展对软件开发有影响。



聚光灯图标表明编程技巧或重点内容，进行更详细的讨论或对当前论题给出边框解释。

## 关于作者

吉姆·考恩 (James P. Cohoon, 爱称 Jim) 是弗吉尼亚大学计算机科学系教授，并曾是 AT&T 公司贝尔实验室的技术人员。他在明尼苏达大学获得 Ph.D. 后，来到弗吉尼亚大学任教。他两次获得系里提名为本校教学优秀奖。1994 年，科亨教授在德国讲授 C++ 和软件工程获得富布莱特奖金。科亨教授的研究兴趣包括算法、电子系统的计算机辅助设计、决策优化和计算机科学教育。他在这些领域发表了 60 多篇论文。他是美国计算机协会 (ACM) 会员、ACM 设计自动化专业组成员 (SIGDA)、ACM 计算机科学教育专业组成员 (SIGCSE)、IEEE 会员及 IEEE 电路与系统学会成员。他还是 ACM 出版与 SIG 委员会成员，并担任过 SIGDA 的主席。他的电子邮件地址是 [cohoon@virginia.edu](mailto:cohoon@virginia.edu)。他的 Web 主页是 <http://www.cs.virginia.edu/~cohoon>。

杰克·戴卫森 (Jack W. Davidson) 也是弗吉尼亚大学计算机科学系教授。他在亚利桑那大学获得 Ph.D. 后，来到弗吉尼亚大学任教。戴维德森教授曾因教学创新而获得 NCR 成员创新奖。戴维德森教授的研究兴趣包括编译、计算机体系结构、系统软件和计算机科学教育。他在这些领域发表了 80 多篇论文。他是 ACM 会员、ACM 编程语言专业组成员 (SIGPLAN)、ACM 体系结构专业组成员 (SIGARCH)、ACM 计算机科学教育专业组成员 (SIGCSE)、IEEE 会员及 IEEE 计算机学会会员。他在 1994 年至 2000 年作过 ACM 重要杂志编程语言与系统学刊的副主编 (Transactions on Programming Languages and Systems)。他曾担任编程语言设计与实现 1998 年会 (PLDI'98) 主席，并担任 2000 年 SIGPLAN 关于语言、编译和嵌入系统研讨会 (LCTES 2000) 的程序委员会副主席。他的电子邮件地址是 [jwd@virginia.edu](mailto:jwd@virginia.edu)。他的 Web 主页是 <http://www.cs.virginia.edu/~jwd>。

## 参考读物

下面是 C++ 语言的主要参考书：

· International Standard for Information Systems——Programming Language C++ ( 信息系统国际标准——C++ 编程语言 )，ISO/IEC FDIS 14882，Washington DC: American National Standards Institute，1998。

· B. Stroustrup, *The C++ Programming Language* ( C++ 编程语言 )，第 3 版，Reading, MA: Addison-Wesley，1998。

下面是关于库与更高级面向对象设计、程序开发和标准模板库的好读物：

· J. Bergin, *Data Abstraction: The Object-Oriented Approach Using C++* ( 数据抽象：使用 C++ 的面向对象方法 )，New York: McGraw-Hill，1994。

· M. D. Carroll 与 M. A. Ellis, *Designing and Coding Reusable C++* ( 可重用 C++ 的设计与

编码), Reading, MA: Addison-Wesley, 1995。

· M. P. Cline 与 G. A. Lomow, *C++ FAQs* (C++ 常见问题集), Reading, MA: Addison-Wesley, 1995。

· A. Koenig 与 B. Moo, *Ruminations on C++* (对 C++ 的反思), Reading, MA: Addison-Wesley, 1997。

· S. B. Lippman 与 J. Lajoie, *C++ Primer* (C++ 入门), 第3版, Reading, MA: Addison-Wesley, 1998。

· S. Maguire, *Writing Solid Code* (编写稳定的代码), Redmond, WA: 微软出版社, 1993。

· S. Meyers, *Effective C++*, (有效的 C++), Reading, MA: Addison-Wesley, 1998。

· S. Meyers, *More Effective C++*, (更有效的 C++), Reading, MA: Addison-Wesley, 1996。

· D. R. Musser 与 A. Saini, *STL Tutorial and Reference Guide* (STL 学习与参考指南), Reading, MA: Addison-Wesley, 1995。

· P. J. Plauger, A. Stepanov, M. Lee 与 D. R. Musser, *The Standard Template Library* (标准模板库), Englewood Cliffs, NJ: Prentice-Hall, 1998。

· B. Stroustrup, *The Design and Evolution of C++* (C++ 的设计与发展), Reading, MA: Addison-Wesley, 1994。

下面是更多学习计算机历史与未来的好读物：

· S. Augarten, *Bit by Bit: An Illustrated History of Computers* (逐位前进：计算机的历史图解), New York: Ticknor & Fields, 1984。

· P. J. Denning 与 B. Metcalfe (编辑), *Beyond Calculation: The Next Fifty Years of Computing* (超越计算：今后 50 年的计算), New York: Copernicus Press, Springer-Verlag, 1997。

· J. A. N. Lee, *Computer Pioneers* (计算机先驱), Piscataway, NJ: IEEE Press, 1995。

· J. Palfreman 与 D. Swade, *The Dream Machine: Exploring the Computer Age* (梦幻机器：探索计算机时代), London: BBS Books, 1991。

· H. G. Stine, *The Untold Story of the Computer Revolution* (计算机革命未曾讲过的故事), New York: Arbor House, 1985。

· M. R. Williams, *A History of Computing Technology* (计算技术的历史), Englewood Cliffs, NJ: Prentice-Hall, 1985。

詹姆斯·考恩 (J.P.C.)  
杰克·戴卫森 (J.W.D.)

# 目 录

第 1 章 计算与面向对象设计方法学 .....	1
1.1 基本计算术语 .....	1
1.2 软件 .....	13
1.3 工程软件 .....	18
1.4 面向对象设计 .....	23
1.5 记忆要点 .....	30
1.6 参考文献 .....	31
1.7 习题 .....	31
第 2 章 C++基础 .....	35
2.1 程序组织 .....	35
2.2 第一个程序 .....	35
2.3 第二个程序 .....	37
2.4 注释 .....	38
2.5 赋值 .....	40
2.6 基本的 C++对象 .....	41
2.7 常数 .....	44
2.8 名字 .....	51
2.9 定义 .....	54
2.10 表达式 .....	57
2.11 输出语句 .....	67
2.12 计算平均速度 .....	69
2.13 记忆要点 .....	72
2.14 习题 .....	74
第 3 章 修改对象 .....	78
3.1 赋值 .....	78
3.2 const 定义 .....	82
3.3 输入语句 .....	83
3.4 计算碳氢化合物的分子数 .....	85
3.5 复合赋值 .....	88
3.6 增量和减量 .....	90
3.7 年度存款的估算 .....	92
3.8 字符串类 .....	94

3.9	EzWindows.....	101
3.10	草地割草.....	105
3.11	记忆要点.....	110
3.12	习题.....	112
<b>第 4 章</b>	<b>控制结构.....</b>	<b>117</b>
4.1	布尔代数.....	117
4.2	布尔类型.....	119
4.3	if 语句的条件执行过程.....	126
4.4	switch 语句的条件执行过程.....	137
4.5	按要求计算表达式.....	141
4.6	判定日期的有效性.....	144
4.7	使用 while 语句的循环.....	150
4.8	简单的串和字符处理.....	156
4.9	使用 for 结构的循环.....	165
4.10	简单的数据可视化.....	172
4.11	解答懒汉谜题.....	174
4.12	使用 do 结构的循环.....	175
4.13	记忆要点.....	178
4.14	习题.....	179
<b>第 5 章</b>	<b>函数基础.....</b>	<b>188</b>
5.1	函数的基本概念.....	188
5.2	预处理器.....	196
5.3	使用软件库.....	199
5.4	iostream 库.....	200
5.5	iomanip 库.....	203
5.6	fstream 库.....	209
5.7	随机数.....	215
5.8	assert 库.....	220
5.9	记忆要点.....	223
5.10	参考文献.....	224
5.11	习题.....	224
<b>第 6 章</b>	<b>程序员定义的函数.....</b>	<b>229</b>
6.1	基本概念.....	229
6.2	一个诱人的问题.....	232
6.3	一些有用的函数.....	237
6.4	二次多项式的积分.....	239
6.5	局部作用域.....	242

6.6	全局作用域.....	245
6.7	引用参数.....	249
6.8	按引用传递对象.....	257
6.9	验证电话访问密码.....	261
6.10	常量参数.....	264
6.11	默认参数.....	267
6.12	函数参数的类型转换.....	269
6.13	函数重载.....	269
6.14	递归函数.....	273
6.15	显示股价区间图.....	278
6.16	记忆要点.....	286
6.17	参考文献.....	288
6.18	习题.....	288
<b>第 7 章</b>	<b>类结构与面向对象的设计.....</b>	<b>305</b>
7.1	程序员定义的数据类型简介.....	305
7.2	RectangleShape 类.....	307
7.3	使用 RectangleShape 类.....	315
7.4	构造函数.....	318
7.5	制作万花筒.....	320
7.6	面向对象的分析与设计.....	325
7.7	记忆要点.....	334
7.8	参考文献.....	334
7.9	习题.....	335
<b>第 8 章</b>	<b>抽象数据类型的实现.....</b>	<b>339</b>
8.1	抽象数据类型简介.....	339
8.2	有理数 Rational ADT 的基础.....	340
8.3	有理数 Rational 类接口描述.....	345
8.4	实现有理数类.....	354
8.5	拷贝构造函数、成员赋值和析构函数.....	361
8.6	一个伪随机整数的 ADT.....	368
8.7	红 - 黄 - 绿游戏.....	374
8.8	记忆要点.....	391
8.9	习题.....	393
<b>第 9 章</b>	<b>列 表.....</b>	<b>398</b>
9.1	命名集合.....	398
9.2	一维数组.....	399
9.3	数组作为参数.....	410