



PEARSON

C++大学教程

——(第六版)(英文版)——

C++ How to Program, Sixth Edition

尽早接触类/对象/面向对象编程

- 类, 对象, 封装
- 继承, 多态性
- 完整的面向对象编程实例研究:
Time类, GradeBook类, Employee类

基础知识

- 历史, 硬件, 软件
- 输入/输出流, 类型, 运算
- 控制语句, 函数
- 数组, vector类模板
- 指针, 引用
- 字符串类, C风格的字符串
- 运算符重载
- 异常处理, 文件
- 位运算及字符操作
- GNU C++调试器与Visual Studio调试器

数据结构

- 递归, 查找, 排序
- 链表, 队列, 堆栈, 树
- 模板
- 标准模板库: 容器, 迭代器和算法

OOD/UML2 ATM实例研究(选学)

- 确定类, 属性, 状态, 活动, 操作, 协作
- 示图: 用例图, 类图, 状态图, 活动图, 通信图, 顺序图

开源的C++库

- 使用Orge进行游戏编程
- Boost C++库与C++的将来

Introducing
Game Programming
and the **Boost Libraries**

VAROOM



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

DEITEL®

[美]

P. J. Deitel
H. M. Deitel

著

C++

HOW TO PROGRAM

SIXTH EDITION

Pearson International Edition

内 容 简 介

本书是一本讲解 C++ 编程的优秀教材,全面介绍了面向对象编程的原理和方法,详细分析了与 C++ 编程有关的技术,具体包括类与对象、控制语句、函数与递归、数组、指针、运算符重载、继承、多态、模板、流输入/输出、异常处理、文件处理、搜索与排序、数据结构、标准模板库等内容,本书的同步学习网站上还包含了更多的扩展内容。全书以“活代码”方式详细分析每个知识要点,是初学者和中高级程序员学习 C++ 编程的理想用书。

本书可作为高等院校相关专业的编程语言教材和 C++ 编程教材,也是软件设计人员学习 C++ 编程的理想读物。

Original edition, entitled C++ HOW TO PROGRAM, SIXTH EDITION, 9780136152507 by P. J. DEITEL and H. M. DEITEL, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2008 Pearson Education, Inc.

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 Pearson Education, Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY copyright ©2011.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution in the People's Republic of China exclusively(except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版专有出版权由 Pearson Education (培生教育出版集团)授予电子工业出版社。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。

本书在中国大陆地区出版,仅限在中国大陆发行。

本书贴有 Pearson Education (培生教育出版集团)激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字: 01-2010-5782

图书在版编目(CIP)数据

C++ 大学教程 = C++ How to Program: 第 6 版: 英文 / (美)戴特尔 (Deitel, P. J.) 著.

北京: 电子工业出版社, 2011.1

国外计算机科学教材系列

ISBN 978-7-121-12197-5

I. ①C… II. ①戴… III. ①C 语言 - 程序设计 - 高等学校 - 教材 - 英文 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 216396 号

策划编辑: 冯小贝

责任编辑: 许菊芳

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787 × 1092 1/16 印张: 67 字数: 1716 千字

印 次: 2011 年 1 月第 1 次印刷

定 价: 128.00 元 (含 CD 光盘 1 张)

凡所购买电子工业出版社的图书有缺损问题, 请向购买书店调换; 若书店售缺, 请与本社发行部联系。联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

ies Page

SafariX Web Books

www.deitel.com/books/SafariX.html

C++ How to Program, 5/E & 6/E

Java How to Program, 6/E & 7/E

Simply C++: An Application-Driven
Tutorial Approach

Simply Visual Basic 2005: An Application-
Driven Tutorial Approach, 2/E

Small C++ How to Program, 5/E

Small Java How to Program, 5/E

Visual Basic 2005 How to Program, 3/E

Visual C# 2005 How to Program, 2/E

To follow the Deitel publishing program, please register for the free *Deitel® Buzz*
Online e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

To communicate with the authors, send e-mail to:

deitel@deitel.com

For information on corporate on-site seminars offered by Deitel & Associates, Inc.
worldwide, visit:

www.deitel.com

or write to

deitel@deitel.com

For continuing updates on Prentice Hall/Deitel publications visit:

www.deitel.com

www.prenhall.com/deitel

www.InformIT.com/deitel

Check out our Resource Centers for valuable web resources that will help you master
C++, other important programming languages and Web 2.0 topics:

www.deitel.com/ResourceCenters.html

C++

HOW TO PROGRAM

SIXTH EDITION

Pearson International Edition

P. J. Deitel

Deitel & Associates, Inc.

H. M. Deitel

Deitel & Associates, Inc.



Upper Saddle River, New Jersey 07458

Trademarks

DEITEL, the double-thumbs-up bug and DIVE INTO are registered trademarks of Deitel and Associates, Inc.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Pearson Education is independent of Sun Microsystems, Inc.

Microsoft, Internet Explorer and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Pong® classic video game courtesy of Atari Interactive, Inc. ©2007 Atari Interactive, Inc. All rights reserved. Used with permission.

UNIX is a registered trademark of The Open Group.

*To the victims of the tragedy at Virginia Tech,
their families, their friends and the
Virginia Tech community.*

Paul and Harvey Deitel

Deitel Resource Centers

Our Resource Centers focus on the vast amounts of free content available online. Start your search here for resources, downloads, tutorials, documentation, books, e-books, journals, articles, blogs, RSS feeds and more on many of today's hottest programming and technology topics. For the most up-to-date list of our Resource Centers, visit:

www.deitel.com/ResourceCenters.html

Let us know what other Resource Centers you'd like to see! Also, please register for the free *Deitel® Buzz Online* e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

Programming	Visual Basic	Java EE 5	Internet Business
.NET	Visual C++	Java SE 6	Affiliate Programs
.NET 3.0	Web Services	JavaFX	Google AdSense
Ajax	Web3D Technologies	JavaScript	Google Analytics
Apex On-Demand	XML		Google Services
Programming		Web 2.0	Internet Advertising
Language	Software	Alert Services	Internet Business
ASP.NET	Apache	Attention Economy	Initiative
C	DotNetNuke (DNN)	Blogging	Internet Public
C#	Eclipse	Building Web	Relations
C++	Firefox	Communities	Link Building
C++ Boost Libraries	Internet Explorer 7	Community	Podcasting
C++ Game	Linux	Generated Content	Search Engine
Programming	MySQL	Google Base	Optimization
Code Search Engines	Open Source	Google Video	Sitemaps
and Code Sites	Search Engines	Google Web Toolkit	Web Analytics
Computer Game	Wikis	Internet Video	Website Monetization
Programming	Windows Vista	Joost	
CSS 2.1		Mashups	Open Source
Flash 9	Microsoft	Microformats	Apache
Flex	.NET	Ning	DotNetNuke (DNN)
Java	.NET 3.0	Recommender	Eclipse
Java Certification and	ASP.NET	Systems	Firefox
Assessment Testing	C#	RSS	Linux
Java Design Patterns	DotNetNuke (DNN)	Skype	MySQL
Java EE 5	Internet Explorer 7	Social Media	Open Source
Java SE 6	Silverlight	Social Networking	Perl
JavaFX	Visual Basic	Software as a Service	PHP
JavaScript	Visual C++	(SaaS)	Python
OpenGL	Windows Vista	Virtual Worlds	Ruby
Perl		Web 2.0	
PHP	Java	Web 3.0	Other Topics
Programming Projects	Java	Widgets	Computer Games
Python	Java Certification and	Wikis	Computing Jobs
Ruby	Assessment Testing		Gadgets and Gizmos
Silverlight	Java Design Patterns		Sudoku

前言

“诚实地对待每个细节，这正是你能力之所在。”

—修女 Teresa

欢迎大家进入 C++ 编程世界，开始学习《C++ 大学教程》(第六版)。在 Deitel & Associates 公司，我们为 Pearson Education (培生教育出版公司) 编写程序设计语言类的教材和专业书籍，提供全球范围的企业培训，并进行 Web 2.0 互联网业务的开发。这本书正是我们的成果之一，它不仅反映了 C++ 语言的发展变化，而且体现了更好的关于编程的教学和学习方法。和前版相比，所有章节都有了显著的调整。此前言的“本书导读”部分将帮助教师、学生和专业人士了解本书所覆盖的 C++ 和面向对象编程知识。

新增特性与更新特性

下面是我们对《C++ 大学教程》第五版和第六版本所做的更新。

- **游戏编程。**新增了一章专门介绍游戏编程。现在计算机游戏产业的收入已经超过了首映电影业务的收入，该产业为有志于从事游戏编程职业的学生创造了许许多多的就业机会。第 22 章“使用 Orge 进行游戏编程”，介绍了如何使用开源的 Ogre 3D 图形引擎软件进行游戏编程和图形绘制。首先讨论游戏编程所涉及的一些基本问题，然后展示如何使用 Ogre 开发一个简单的游戏，它的玩法与一款最初由 Atari 在 1972 年开发的经典视频游戏 Pong 相似。我们将演示如何创建一个三维彩色图形场景、平滑绘制移动的物体、使用计时器来控制动画的速度、检测物体之间的碰撞、添加声效、接受键盘输入和显示文本输出等。
- **C++ 的将来。**增加了第 23 章来考虑未来的 C++——介绍了 Boost C++ 库，技术报告 1 (TR1) 和 C++0x。这个免费的 Boost 开源库是由 C++ 社区 (C++ community) 的成员开发的。TR1 描述了对 C++ 标准库的修改建议，其中很多是基于当前的 Boost 库的。C++ 标准委员会正在修订 C++ 标准。新标准的主要目标是令 C++ 更容易学习，提高库的构建能力，并增强与 C 程序设计语言的兼容性。最近的标准是在 1998 年公布的。目前称为 C++0x 的新标准化工作始于 2003 年。除了对核心语言的变更外，新标准极可能包括 TR1 中的许多库。我们对 Boost 库进行概述，并提供有关“正则表达式”和“智能指针”库的代码示例。正则表达式用于在文本中匹配特定的字符模式，它们可以用于验证数据以保证它具有特定的格式，可以用于用另外的字符串替换一个字符串中的一部分，可以用于拆分字符串等。C 和 C++ 代码中的许多常见错误都是与指针有关的。指针是第 8 章中学习的一种强大的编程能力，而智能指针将通过对标准指针提供附加的功能来帮助避免一些错误。
- **大幅度的内容修订。**所有章节都经过了有意义的校正和提升。我们不断改进，使语言表达更为清晰与准确。同时，也调整了 C++ 术语的表达方式，使其与定义该语言的 ANSI/ISO C++ 标准文档相适应。
- **尽早接触类和对象。**本书在第 1 章介绍对象技术的基本概念和术语，在第 3 章学生即开始开发自定义的、可重用的类及对象。无论从书的开篇，还是贯穿整本书，只要适合，这本书都在体现面向对象编程。尽早地讨论类和对象可以使学生直接“考虑对象”和更彻底地掌握这些概念。面向对象编程不是一件简单的事情，但是用面向对象的思想编写程序是一种乐趣，相信学生很快就能体会到。
- **综合的实例研究。**书中提供了若干横跨多个章节的实例研究，它们往往都基于本书前面引入的类，然后在后面的章节中继续阐述新的编程概念。这些实例研究包括第 3~7 章中 GradeBook (成绩簿) 类的开发，第 9~10 章中 Time (时间) 类的开发，第 12~13 章中 Employee (雇员) 类的开发，以及既出现在第 1~7 章、第 9 章和第 13 章，又出现在附录 G 的可选学的 OOD/UML ATM (自动取款机) 实例研究。
- **完整的 GradeBook 实例研究。**GradeBook 实例研究不断增强早期提出的类，它用第 3~7 章中的类和对象一步一步地创建 GradeBook 类。GradeBook 类表示了教师的成绩簿，并且可以根据一群学生的成绩执行各种计算，例如计算平均成绩，找出最高和最低的成绩，以及打印条形图等。

- **统一建模语言 2 (UML 2)**。统一建模语言已经成为面向对象系统设计者首选的图形建模语言。本书中所有的UML图示都符合UML 2规范。我们使用UML类图可视化地表示出类以及它们之间的继承关系,使用UML活动图展示各个C++控制语句中的控制流。尤其是在可选学的OOD/UML ATM实例研究中,大量使用了UML。
- **可选学的OOD/UML ATM实例研究**。这个可选学的OOD/UML ATM实例研究出现在第1~7章、第9章和第13章的软件工程实例研究部分,非常适合于第一门和第二门程序设计课程的教学。组成该实例研究的9个章节循序渐进地介绍如何采用UML语言进行面向对象的设计。首先介绍UML 2的一个精简子集,然后指导大家体验第一项设计,这项设计是专为面向对象设计或编程的新手而准备的。这一实例研究的目标是通过逐步展开面向对象的设计,帮助学生增强在第1章开始学习并在第3章实现的面向对象编程的概念。这个实例研究已通过一个精通OOD/UML的杰出团队的审查,团队的专家来自学术界和工业界。而且,该实例研究并不是一个简单意义上的练习,更确切地说,应该是一个精心设计的、一环扣一环的学习过程。此过程通过对一个完整的877行C++代码实现的详尽诠释而结束。后面将对这个实例研究的9个部分进行详细的介绍。
- **多个源文件程序的编译和连接过程**。第3章包含了编译和连接过程的详细图示和讨论,该过程生成一个可执行的应用程序。
- **函数调用栈的解释**。第6章通过提供函数调用栈和活动记录的详细讨论(含有图示),来解释C++是如何记录正在运行的函数,内存是如何保存函数的自动变量,以及函数在完成它执行的任务后是如何知道返回何处的。
- **C++标准库中的string和vector对象**。通过使用string类和vector类,可以在书中较前面的示例程序中更多地体现面向对象的性质。
- **string类**。在整本书中,对于大多数字符串的操作都使用string类,而非C风格的基于指针的char*字符串。但是,在第8章、第10章和第11章中我们仍讨论char*字符串的有关内容。这样,给学生练习指针操作的机会,举例说明如何用new和delete进行动态内存分配,构建自己的String类,并使学生今后有能力应对C和C++遗留代码中的char*字符串问题。
- **类模板vector**。整本书都使用类模板vector,而不是C风格的基于指针的数组操作。尽管如此,仍会在第7章中继续讨论C风格的基于指针的数组,为学生以后能够处理遗留的C和C++代码做准备,同时也为第11章中建立自定义的Array类奠定基础。
- **继承和多态性的协调处理**。第12章和第13章经过仔细调整,使用Employee类层次结构来介绍继承和多态性,使得关于继承和多态性的处理更清楚,更容易被刚刚接触OOP的学生所接受。
- **对多态性工作“知其然也知其所以然”的讨论和图示**。第13章通过详细的图示和说明,解释了C++内部是如何实现多态性、虚函数和动态绑定的,使学生对这些机制的实际工作原理有非常深刻的理解。更重要的是,它有助于学生认识到实现多态性是要付出代价的,即需要额外占用内存和处理器时间,从而使他们能够决定什么时候应该使用多态性,什么时候应避免使用。
- **标准模板库(STL)**。如果基于软件重用性的观点,这一部分应该是本书中最重要的章节了。STL定义了功能强大的、基于模板的和可重用的组件,这些组件实现了许多常见的数据结构及用于处理这些数据结构的算法。第21章介绍了STL,并讨论它的三个关键组件:容器、迭代器和算法。我们展示了使用STL组件可以提供极强大的表达能力,并可以将多行的程序代码简化为单条语句。
- **遵循ANSI/ISO C++标准**。我们根据最新的ANSI/ISO C++标准文档审核了相关的文字表述,以实现完整性和精确性。[注意:如果读者需要更进一步地了解C++技术的细节内容,请阅读C++标准文档。C++标准文档有一份PDF格式的电子版本(文档序列号为INCITS/ISO/IEC 14882-2003),可以从webstore.ansi.org/ansidocstore/default.asp购买。]
- **调试器附录**。本书中放入了两个“使用调试器”的附录——附录I和附录J。
- **多平台的代码测试**。我们在各种流行的C++平台上测试了本书的代码实例。对大多数情况而言,书中所有例子在主流的符合标准的编译器上都能成功运行。
- **不同平台上的错误和警告**。对于一些故意设置错误来阐述关键概念的程序,我们列出了不同操作平台上的错误信息。

上述的一切内容都已经过学术界和工业界著名专家的细致评审，他们与我们就《C++ 大学教程》第五版和第六版的工作进行了意义深远的合作。

我们相信本书及其辅助材料将为学生和专业人员提供一次集知识性、趣味性、挑战性和娱乐性于一体的 C++ 教育体验。书中包括了一整套教辅材料，帮助教师最大限度地发挥学生的学习能力。

在阅读本书的过程中，如果有任何疑问，请发送电子邮件到 deitel@deitel.com，我们将会尽快回复。为了获取本书的更新和所有 C++ 支持软件的具体情况，以及关于 Deitel 出版物和服务的最新资讯，请访问 www.deitel.com。请赶快在 www.deitel.com/newsletter/subscribe.html 上注册免费的 DEITEL Buzz Online 电子邮件时事通信，并经常在 www.deitel.com/ResourceCenters.html 查看所列的越来越多的 C++ 及相关资源中心。每周我们都会在时事通信中公布最新的资源中心。同时，也请告诉我们您所感兴趣的其他资源中心。

教学方法

《C++ 大学教程》(第六版) 含有大量的实例。这本书着重考虑良好的软件工程原则，并强调程序的清晰性。我们通过实例来教学。作为教育工作者，我们在全球范围内讲授工业界最前沿的程序设计语言和软件相关知识。Harvey M. Deitel 博士有着 20 年的高校教学经验和 19 年的工业界教学经历。另一作者 Paul J. Deitel 也具有 16 年的工业界教学资历。他们针对来自政府机构、工业界、军事部门和学术界的 Deitel & Associates 公司的不同客户，从不同的层次讲授 C++ 课程。

采用活代码方式。《C++ 大学教程》(第六版) 拥有大量的“活代码”实例。通过这种方式，对于每个新概念，都用完整的、能实际运行的 C++ 应用程序进行介绍，程序代码之后直接附有一个或者多个运行示例，用于展示程序的输入/输出。这种风格体现了我们讲授和编写程序的方法，称为“活代码”(Live-Code) 方式。

语法的颜色标记。与大多数 C++ 集成开发环境或代码编辑器类似，我们对所有的 C++ 代码都通过不同颜色来标记其中的语法。这样，大大提高了代码的可读性——本书提供的完整的、可执行的 C++ 程序拥有 20 000 行代码，可读性的确是一个特别重要的目标。进行语法颜色标记的约定如下^①：

注释采用绿色显示
关键字采用深蓝色显示
错误采用红色显示
常量和字符串采用浅蓝色显示
其他所有的代码采用黑色显示

代码高亮显示。每个程序中的关键代码段用灰色的矩形块标记。

使用不同的字体和颜色突出重点。我们对关键的术语以及索引页中术语定义出现的地方都用加粗的蓝色字体来表示，从而方便查找；用加粗的 Helvetica 字体来强调是屏幕上出现的元素（例如 File 菜单），而用 Lucida 字体来显示 C++ 程序的内容（例如，`int x = 5;`）。

Web 访问。《C++ 大学教程》(第六版) 的所有源代码实例都可以从下面的地址下载：

<http://int.prenhall.com/deitel>

下载所有的实例后，请在阅读相应的教材内容时运行一下每个程序。然后可以对这些例子做一些改动，通过程序的运行马上看到改动的效果，这是增强学习 C++ 效果的一种立竿见影的好方法。

学习目标。在每章的开始部分列出了一系列学习目标，目的是使大家知道每章的学习主题，并且在学完每章后，有机会确定自己是否到达预期的目标。

名人名言。在每章的学习目标之前，我们引用了一些名人名言。有些是幽默的，有些富有哲理，有些则提供了有趣的见解。希望读者能够细细品味它们所蕴含的与章节内容相关的深层意义。

提纲。每章的提纲有助于大家以自上而下的方式学习这些内容。这样可以预见到即将学习什么，进而轻松、有效地调整学习节奏。

插图/图表。本书包含了大量的图表、表格、线条图、程序及程序输出。我们采用 UML 活动图表示控制语句中的控制流，用 UML 类图表示类的数据成员、构造函数和成员函数。在可选学的 OOD/UML ATM 实例研究中，更是广泛使用了 6 种主要的 UML 图。

^① 由于印刷原因，英文影印版中取消了颜色区分——编者著。

编程提示。本书还包含了很多编程提示，目的在于帮助学生将学习重点放在程序开发的关键部分。这些提示和实践经验是两位作者累积60年的编程和教学经验总结。我的一位数学专业的学生说，这种方法就像在数学中着重标明公理、定理、引理和推论一样，同样为编写一个好的软件提供了基础。



良好的编程习惯

良好的编程习惯关注的技术将有助于编写出更加清晰、更易理解和更易维护的程序。



常见的编程错误

学生总是犯一些常见的错误。指出这些常见的编程错误可以减少犯类似错误的可能性。



错误预防技巧

这些提示包含发现和消除程序错误的各种建议。事实上，许多这样的提示描写的是如何在开始编写C++程序的第一时间就防止错误。



性能提示

学生喜欢使他们的程序“超值”(turbo charge)，快速而高效。这些提示提供了一些强调提高程序性能的方法，使程序运行得更快，或者它们所占用的内存最少。



可移植性提示

书中所包含的可移植性提示将有助于编写出可以运行在不同平台上的代码，同时也使大家明白C++是如何实现其高度可移植性的。



软件工程知识

软件工程知识这一部分的提示突出了关于软件体系结构和设计的问题，这些问题往往影响到整个软件系统的构建，特别是对一些大型的系统而言。

摘要列表。每章都是以附加的教学内容作为结尾的。本书的一个教学特色，是在每章结束部分都以清单的形式逐节汇总了本章要点。

术语。在这一部分提供了一个按字母顺序排列的术语清单，列出了每章中所定义的重要术语。

自测题及答案。每章都包含了为自学而精心设计的大量自测题及其答案。

练习题。每章最后都提供了大量的练习题，包括：

- 对一些重要术语和概念的简单回顾
- 在代码实例中找出错误
- 编写单条的C++语句
- 编写小块的C++函数和类
- 编写完整的C++的函数、类和程序
- 开发大型的课程项目

如此大量的练习题有助于教师根据学生的特殊需要来调整课程，以及在不同的学期布置不同的作业等。教师可以利用这些题目来安排家庭作业、小测验、大型考试和课程项目。如果需要更多的其他可能的练习和项目资源，可以访问我们的编程项目资源中心 (www.deitel.com/ProgrammingProjects/)。

注意：请不要给我们写信，要求访问 Pearson Education 教师资源中心。只有采用这本教材进行授课的高校教师才可以访问该中心。教师只能通过 Pearson Education 的代理商获得访问资格^①。

成千条的索引项。本书包含了规模庞大的索引。索引是非常有用的，尤其当你以此书作为参考书的时候。

本书导读

这一部分将带领大家了解一下在《C++ 大学教程》(第六版)中可以学到的许多C++功能。图1给出了各个章节之间的依赖关系。我们建议沿着箭头所指的顺序来学习各个主题，当然，也可以按照其他的路线。这本书可广泛适用于各个层次的C++课程授课中。在网络上搜索“syllabus”、“C++”和“Deitel”等关键字，会找到不少关于这本书的最新版本的教学大纲。

^① 国内教师申请教辅资源的方法，请参见本书末尾的“教学支持说明”页。

第3章：类和对象介绍，很自然地将类和对象较早地引见给大家。这一章有助于学生从一开始就顺利接受面向对象的思想。这部分工作是在一支优秀的评审队伍的指导下完成的，这支队伍由工业界和学术界著名的专家组成。我们通过一系列简单的实例来介绍类、对象、成员函数、构造函数和数据成员。在本章中，我们探索出了一个很好的工程化的框架，用来组织用C++编写的面向对象程序。首先，用一个简单实例来引出类的概念。然后，通过精心策划的7个循序渐进可执行的完整程序来展示如何创建和使用自己定义的类。这些例子从关于成绩簿类的开发这一完整实例研究作为出发点，教师可以使用成绩簿类来维护学生的考试成绩。在后续的几章中不断增强这个实例研究，最终版本在第7章中得以实现。GradeBook类实例研究描述了如何定义一个类，以及如何使用它去创建对象；讨论了如何声明和定义成员函数以实现类的一些行为，如何声明数据成员来实现类的属性，以及如何调用对象的成员函数使它们执行任务。在这一章，还介绍了C++标准库中的类string，并创建string对象来保存GradeBook对象所表示的课程的名字；解释了类的数据成员和函数的局部变量之间的差别，以及如何使用构造函数来保证对象的数据在对象创建时就被初始化；说明了通过将类定义和使用类的客户代码（如函数main）相分离，可以提高软件的可重用性。第3章也介绍了另一条基本的软件工程原则——将接口从实现中分离出来，并包含了一张详细的示意图和相应的讨论，解释了产生可执行程序的编译和连接过程。

第4章：控制语句（第I部分），重点在于创建有用的类的程序开发过程。这一章讨论了如何面对问题陈述，继而开发出一个可解决问题的C++程序，这当然包括用伪代码实现的那些中间步骤。本章介绍了一些简单的控制语句，例如用于判断的if和if...else语句和用于循环的while语句。我们利用第3章的GradeBook类来考究计数器控制的循环和标记控制的循环，并介绍了C++的自增、自减和赋值运算符。这一章包含GradeBook类的两个增强版，每个都基于第3章的最后版本。这些版本各自都包含了一个用控制语句计算一组学生成绩平均分的成员函数。第一个版本中的成员函数采用计数器控制的循环，由用户输入10名学生的成绩，然后计算出平均分。第二个版本的成员函数则采用标记控制的循环，由用户输入任意多名学生的成绩，然后计算平均分。该章还使用简单的UML活动图来呈现通过每个控制语句的控制流。

第5章：控制语句（第II部分），继续以实例的方式讨论C++中的控制语句，包括for循环语句、do...while循环语句、switch选择语句、break语句和continue语句。我们创建了GradeBook类的一个增强版本，它使用switch语句对用户输入的A、B、C、D和F五个等级的成绩个数分别进行统计。这个版本采用标记控制循环来输入成绩。在从用户那里读入成绩的同时，一个成员函数就在修改记录相应等级成绩个数的特定数据成员。之后，GradeBook类的另一个成员函数使用这些数据成员，显示一份基于这些输入成绩的汇总报告。这一章还进行了逻辑运算符的讨论。

第6章：函数和递归入门，深入了解对象及其成员函数。我们讨论了C++标准库函数，并进一步探讨了学生如何构建自己的函数的问题。在第6章中阐述的技术对于编写正确组织的程序是至关重要的，尤其是那些在现实世界应用中很可能由系统程序员和应用程序员开发的大型程序或软件。“分而治之”策略是解决复杂问题的有效手段，其做法是将一个复杂的问题分解成较简单的、相互作用的部分。本章的第一个例子继续进行GradeBook类实例研究，其中列举的函数具有多个参数。学生将会喜欢这一章中关于随机数和模拟方面的知识，以及关于掷骰子游戏的讨论。在解决这些问题的时候，恰到好处地应用了控制语句。本章讨论了所谓的“C++是更好的C”的话题，其中包括内联函数、引用参数、默认实参、一元作用域分辨运算符、函数重载和函数模板等。还介绍了C++的按值调用和按引用调用的功能。头文件表介绍了很多头文件，这些文件是大家在整本书中经常用到的。在这个新的版本中，通过对函数调用堆栈和活动记录的详细讨论（含有图示），解释了C++如何能够跟踪正在运行的当前函数，函数的自动变量如何在内存中进行维护，以及函数在完成它执行的任务后如何知道返回何处。这一章为递归的学习奠定了扎实的基础，并提供了一个表格，总结了会在本书其余部分介绍的大量的递归例子和练习题。有的教材喜欢将递归安排在最后一章，但我们认为这个主题最好均匀地分布于全书各处进行讲解。在本章末尾的大量练习题中，包括了几个经典的递归问题，如Hanoi塔问题等。

第7章：数组与vector，解释如何处理值的列表和表格。我们讨论了将数据组织为具有相同类型的数据元素的数组问题，并表明数组使对象执行其任务更方便。本章前面的部分仍然使用C风格的基于指针的数组。这样的数组，正如大家将在第8章中所看到的，可以视为是指向内存中数组内容的指针。在这一章数组还会以成熟的对象形式出现，我们将介绍C++标准库中的vector类模板，它是一个具有很强健壮性的数组数据结构。本章列举了许多一维数组和二维数组的例子。这章中的例程涉及各种常见的数组操作、直方图打印、数据排序和将数组传给函数等。在本章中有两节含有最后一些关于GradeBook类实例研究的内容。在这两节中，我们在程

序执行期间使用数组来保存学生的成绩。该类的以前版本只是对用户输入的一系列成绩进行处理，并没有在类的数据成员里保存这些成绩。这次则利用数组使 GradeBook 类的对象在内存中维护这些成绩，从而省去了重复输入这些成绩的过程。本章中 GradeBook 类的第一个版本，利用一个一维数组来存储这些成绩，并且可以产生一份报告，其中包含了这些成绩的平均分、最高分、最低分以及一张表示成绩分布的直方图。本章中 GradeBook 类的第二个版本（也是关于这个实例研究的最后一个版本），则用一个二维数组来存储很多学生一学期里多门考试的成绩。这个版本既可以计算单个学生的学期平均分，又可以得到整个学期所有成绩的最高分和最低分。当然该版本的类也可以产生一张关于这个学期整体成绩分布的直方图。这一章的另一个重要特色就是对基本的排序和搜索技术的讨论。该章后面的练习题包括了大量有趣且富有挑战性的问题，例如改进的排序技术、一个简单的航班订票系统的设计、海龟图概念入门（因 LOGO 语言而著名）、骑士周游和八皇后问题等，这些都涉及人工智能领域广泛使用的启发性编程的概念。这些练习题总结了许多递归问题，包括选择排序、回文、线性搜索、八皇后问题、数组打印、逆序打印字符串及在数组中查找最小值等。

第8章：指针和基于指针的字符串，介绍 C++ 语言中功能最强大的特性之一——指针。这一章详细解释了指针运算符、按引用调用、指针表达式、指针算术运算、指针和数组之间的关系、指针数组和函数指针等。我们通过讲授指针和 const 一起使用的方法，实践以最少的特权建立更健壮的软件这一原则。讨论了使用 sizeof 运算符来确定在程序编译期间数据类型或者数据项的字节大小。在 C++ 中，指针、数组和 C 风格的字符串之间有着密切的联系。因此，我们介绍了基本的 C 风格的字符串操作的概念，并讨论了一些最常用的 C 风格字符串处理函数，例如 getline（输入一行文本）、strcpy 和 strncpy（复制一个字符串）、strcat 和 strncat（连接两个字符串）、strcmp 和 strncmp（比较两个字符串）、strtok（将一个字符串“标记”为不同的部分）和 strlen（返回一个字符串的长度）等。在这个新版教材中，我们尽量用 string 对象（在第3章中已介绍）来代替 C 风格的 char * 基于指针的字符串。但是，第8章中仍然使用 char * 字符串，目的在于帮助读者掌握指针，并为将来的职业生涯做好准备。因为在以后的工作中，可能会遇到大量三十多年来实现的 C 遗留代码。所以，大家非常有必要熟悉 C++ 中这两种最盛行的创建和操作字符串的方法。许多人发现，迄今为止在基础的程序设计课程中，指针这部分内容无疑是最难学的。在 C 和“原始 C++”中，数组和字符串其实就是指指针，它们指向内存中数组和字符串的内容（甚至函数的名字也是指针）。通过这一章的认真学习，大家能深入地理解指针这一主题。此章配有富有挑战性的练习题。练习题包括对经典龟兔赛跑问题的模拟、洗牌和发牌问题的算法、递归快速排序和递归走迷宫等。另外，还包括一个特殊的部分，命名为“建立自己的计算机”。这个部分解释了机器语言的编程，并实际开发一个项目，目的是设计和实现一个计算机模拟器，让学生编写和运行机器语言程序。本教材的这个独一无二的特色，对于那些想要知道计算机的真正工作原理的读者是特别有用的。事实上，我们的学生很喜欢这个项目，并经常做出实质性的改进，其中许多的改进已体现在练习题中。另一个很特别的部分包含了颇具挑战性的字符串操作练习题。这些练习题涉及文本分析、字处理、以各种格式打印日期、支票保护、写入支票金额的大写形式、摩尔斯编码和公制 - 英制单位转换等。

第9章：类的深入剖析（第 I 部分），继续讨论面向对象的编程。本章通过内容充实的 Time 类的实例研究来阐述类成员的访问、接口与实现的分离、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、通过默认的“逐个成员复制”进行赋值以及软件的重用性等内容。学生将明白在对象的生命周期中构造函数和析构函数的调用顺序。Time 类实例研究的另一个版本向大家演示了在成员函数返回对 private（私有）数据成员的引用时所产生的问题，它破坏了类的封装性。这一章的练习题向学生提出了挑战，要求开发关于 Time（时间）、Date（日期）、Rectangle（矩形）和游戏 TicTacToe（三连棋）的类。通常，学生都很喜欢玩游戏的程序。那些爱好数学的读者会更喜欢创建诸如 Complex 类（针对复数）、Rational 类（针对有理数）和 HugeInteger 类（针对任意大的整数）等的练习题。

第10章：类的深入剖析（第 II 部分），继续学习类并讲解其他的面向对象编程概念。这一章讨论了常量对象的声明和使用、常量成员函数、组成（composition）（指构建类的过程，它用其他类的对象作为自己的成员）、friend（友元）函数和 friend 类（它们具有访问类的 private 和 protected 成员的特权）、this 指针（它允许对象知道自己的地址是什么）、动态内存分配、static（静态）类成员（用于保存和操作类范围内的数据）、常用的抽象数据类型的例子（数组、字符串和队列）、容器类和迭代器等。在 const 对象的讨论中，我们提到了关键字 mutable。它的用法非常微妙，允许对 const 对象中的“不可见”的实现进行修改。讨论了使用 new 和 delete 进行动态内存分配。当执行 new 失败时，程序会按照默认的方式中止，因为在标准的 C++ 中 new “抛出一个异常”。为了展

开对static类成员的讨论,本章给大家展示了一个基于视频游戏的实例。我们强调了在类的客户面前隐藏类的实现细节的重要性。为此,接下来讨论代理类。代理类是在类的客户面前隐藏实现(包括类头文件中的private数据)的有效手段。这一章的练习题包括了开发一个储蓄存款账户的类和一个拥有整数集合的类。

第11章:运算符重载: String和Array对象,介绍C++课程中最受欢迎的主题之一。学生的确对这一章的内容抱有浓厚的兴趣。他们发现,本章是对“精雕细琢”深入讨论类的第9章和第10章内容的完美补充。通过运算符重载(operator overloading),程序员可以告诉编译器如何将现有的运算符应用于新类型的对象。C++已经知道如何将运算符用于内置类型的对象,例如整数、浮点型数据和字符型数据。但是,假如我们新创建了一个类String,那么在String对象之间使用加号会是什么意思呢?很多程序员对字符串使用加号(+),意思是连接相邻的两个字符串。在第11章中,大家将学习如何重载加号。这样,当一个表达式中的两个String对象之间书写了加号时,编译器将产生对一个“运算符函数”的函数调用,该函数将连接这两个String对象。这一章讨论了运算符重载的基本知识、运算符重载时的限制、类成员函数重载和非成员函数重载的比较、一元和二元运算符重载及不同类型间的转换等。第11章的一个特点是收集了许多有意义的实例研究,包括数组类、String类、日期类、大整数类和复数类(最后这两个类及其完整源代码出现在练习题中)。喜欢数学的学生应该会喜欢创建polynomial(多项式)类的练习题。这里材料与大多数的程序设计语言和课程不同。运算符是一个复杂的主题,但也是一个丰富多彩的主题。合理地使用运算符重载会使您的类“锦上添花”。对于已在使用C++标准库中的string类和vector类模板的学生来说,对具有类似功能的Array类和String类的讨论显得特别有价值。这一章的练习题鼓励学生为Complex类、Rational类和HugeInteger类添加运算符重载功能,这样可以方便地使用运算符符号操作这些类的对象,就如同在数学中一样,而不是像第10章中的练习题那样要求采用函数调用。

第12章:面向对象编程:继承,介绍面向对象程序设计语言最基本的性能之一——继承(inheritance)。继承是一种软件重用形式。通过继承,程序员可以吸纳现有类的性能并适当增加一些新的性能,从而快速、较容易地开发新的类。在Employee类层次结构的实例研究中,经过充分修订过的这一章通过连续的5个例子,示范了private数据和protected数据的继承,以及有关继承的良好软件工程等内容。首先,以一个具有private数据成员和public(公有)成员函数的类作为示范的开始,这些成员函数操纵类的这些数据。接下来,实现了具有附加性能的第二个类,故意且乏味地复制第一个例子的大多数代码。第三个例子开始涉及继承和软件重用的知识,采用第一个例子中的类作为基类,快速且简单地在新派生的类中继承了基类的数据和函数。这个例子介绍了继承的机制,并说明了派生类不能直接访问其基类的private成员。这样就促使我们给出第四个例子,在其基类中引入了protected数据,并示范了派生类确实可以访问从基类继承的protected数据。最后一个例子详细说明了正确的软件工程,它将基类的数据定义为private,使用基类的public成员函数(被派生类所继承)在派生类中操纵基类的private数据。本章讨论了基类和派生类、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,以及与继承有关的软件工程等。在这一章中,还比较了继承(“is-a”关系)与组成(“has-a”关系),并介绍了“uses-a”和“knows-a”关系。

第13章:面向对象编程:多态性,介绍面向对象编程的另一种基本性能,即多态性行为。全面修订的第13章基于第12章讲授的继承概念,关注的是类层次结构中各个类之间的关系及这些关系所形成的强大处理能力。当许多类通过继承而与一个公共的类相关时,每一个派生类对象都可视为一个基类对象。这样一来,程序可以按照一种简单而一般的方式,即独立于派生类对象的具体类型来编写。新的类型的对象可以被同一个程序处理,从而使系统更具扩展性。多态性使程序能够消除复杂的switch逻辑,取而代之的是更简单的“直线”逻辑。例如,一个视频游戏的屏幕管理器可以发送一条draw(绘图)消息给绘制对象链表中的每个对象。每个对象都知道如何去绘制自己。只要一个新类的对象知道如何绘制自己,那么它可在不改变程序的情况下直接加入程序。这一章讨论了如何利用virtual(虚)函数实现多态性行为,区分了抽象类(由它不能实例化对象)和具体类(由它可以实例化对象)。抽象类适合为遍布层次结构中的各个类提供一个可继承的接口。通过回顾第12章的Employee类层次结构,举例说明了抽象类和多态性行为。引入了一个抽象基类Employee,而CommissionEmployee类、HourlyEmployee类和SalariedEmployee类直接继承Employee类,BasePlusCommissionEmployee类间接继承Employee类。在过去,我们的专业客户都坚持要我们提供更深层次的解释,能够更清楚地说明在C++中多态性是如何实现的,以及当用这个功能强大的特性编程时执行时间和内存“开销”的情况。为此,我们给出了一张示意图,明明白白地解释了C++编译器为支持多态性而自动构建的vtable(virtual函数表)。最后,介绍了运行时类型信息(run-time type information, RTTI)和动态强制类型转换,它们使程序能够在运行时确定

一个对象的类型，然后对该对象做相应的操作。利用 RTTI 和动态强制类型转换，我们给某个特定类型的雇员增加了 10% 的工资，然后计算这些雇员的所得。而对于其他的雇员类型，则可以通过多态性来计算他们的所得。

第 14 章：模板，讨论 C++ 在软件重用方面比较强大的特色之一，也就是模板（template）。利用函数模板和类模板，编程者可以用一段统一的代码，指定一整套相关的重载函数（这些函数称为函数模板特化）或一整套相关的类（这些类称为类模板特化）。这种技术称为泛型编程（generic programming）。函数模板在第 6 章中做了介绍，在这一章对此继续进行讨论，并给出相应的例子。我们可以为堆栈类编写单一类模板，然后让 C++ 生成各个类模板特化，例如“int 类型的堆栈”类、“float 类型的堆栈”类、“string 类型的堆栈”类，等等。本章讨论如何对类模板使用类型参数、非类型参数和默认类型，还探讨了模板与其他 C++ 的特性（如重载、继承、友元、static 成员等）之间的关系。这一章的练习题很具挑战性，要求学生编写各种各样的函数模板和类模板，并在完整的程序中应用它们。此外，在第 21 章对标准模板库（STL）的容器、迭代器和算法进行讨论时，将再次强化对模板的理解和应用。

第 15 章：输入/输出流，全面讨论标准 C++ 的输入/输出性能。这一章既讨论了足够保证大部分常见的输入/输出操作执行的一系列功能，又概括了其余的功能。许多输入/输出的特性都是面向对象的。输入/输出的这种风格利用了其他 C++ 特性，例如引用、函数重载和运算符重载等。C++ 的各种输入/输出能力包括：用流插入运算符输出、用流提取运算符输入、类型安全的输入/输出、有格式的输入/输出和无格式的输入/输出（出于对性能的考虑）等。用户可以通过重载流插入运算符（<<）和流提取运算符（>>），对自定义类型的对象明确规定如何进行输入/输出。这种可扩展性是 C++ 最有价值的特色之一。C++ 提供了各种流操纵符，可以执行格式化的任务。本章讨论的流操纵符，能够提供诸如以各种进制显示整数、控制浮点数的精度、设置域宽、显示小数点和尾随零、对齐输出、设置和取消格式状态和设置填充字符等功能。此外，这一章还给出了一个创建用户自定义的输出流操纵符的例子。

第 16 章：异常处理，讨论程序员如何通过异常处理，能够编写出鲁棒性强、容错性好并且适用于关键业务和关键任务环境的程序。这一章讨论异常处理适用于何时，介绍具有 try 语句块、throw 语句和 catch 处理器的异常处理的基本性能，指出怎样及何时重新抛出异常，解释如何编写异常说明和处理未预料到的异常，以及讨论异常与构造函数、析构函数和继承之间的重要关系。本章中的练习题向学生展示 C++ 异常处理能力的多样性和强大功能。我们讨论重新抛出异常，并阐明当内存用尽时 new 如何会失败。当 new 失败的时候，许多早期的 C++ 编译器返回默认的 0 值。我们通过抛出一个 bad_alloc（坏分配）异常，展示 new 操作失败的另一种新形式。这一章还演示了如何使用 set_new_handler 函数，指定一个要被调用的定制函数，该定制函数将处理有关内存耗尽的情况；讨论了如何使用 auto_ptr 类模板，隐式地删除动态分配的内存，从而避免内存泄漏。本章最后给出标准库的异常层次结构。

第 17 章：文件处理，讨论创建和处理顺序文件和随机存取文件的技术。这一章首先介绍数据的层次结构，从位到字节、到字段、再到记录，最后至文件。接着，说明从 C++ 的角度如何看待文件和流。我们讨论顺序文件，并构建程序来展示如何打开和关闭文件，如何在文件中顺序地存储数据，以及如何从文件中顺序地读取数据。然后，讨论随机存取文件，并构建程序来说明如何创建随机存取的文件，如何随机地对文件进行数据的读和写，以及如何从随机存取的文件中顺序地读取数据。本章的实例研究将顺序和随机地存取文件的技术结合到一个完整的事务处理程序中。参加我们举办的企业研讨会的学生提到，在学习了关于文件处理的这部分内容后，他们具备了编写实际有价值的文件处理程序的能力，这些程序在他们的企业能够立即得到应用。这一章的练习题要求学生实现各种各样的程序，它们可以构建与处理随机文件和顺序文件。

第 18 章：string 类和字符串流处理，讨论 C++ 由内存的字符串输入数据和向内存的字符串输出数据的能力。这些能力通常称为核心格式化（in-core formatting）或者字符串流处理。string 类是标准库中必不可少的组成成分。出于以下几点原因，在第 8 章及后续章节中继续进行关于 C 风格的基于指针的字符串讨论。第一，这部分内容可以增强读者对指针的理解；第二，在未来十年左右，C++ 编程者必须有能力对近三十年积累起来的大量 C 遗留代码进行阅读和修改。这些遗留代码和工业界多年来已编写的大部分代码一样，都将字符串作为指针来处理。第 18 章讨论 string 对象的赋值、连接和比较，说明如何确定各种各样 string 对象的特性，例如 string 对象的大小、容量、是否为空等，讨论如何重新设置 string 对象的大小。我们考虑各种不同的“查找”函数，它们能够在 string 对象中查找某个子字符串（向前或者向后搜索 string 对象）展示了对于一个选自于字符组成的 string 对象中的字符，如何查找它在 string 对象中的第一次出现或最后一次出现，以及对非出自于 string 对象中的一个