国外计算机科学教材系列

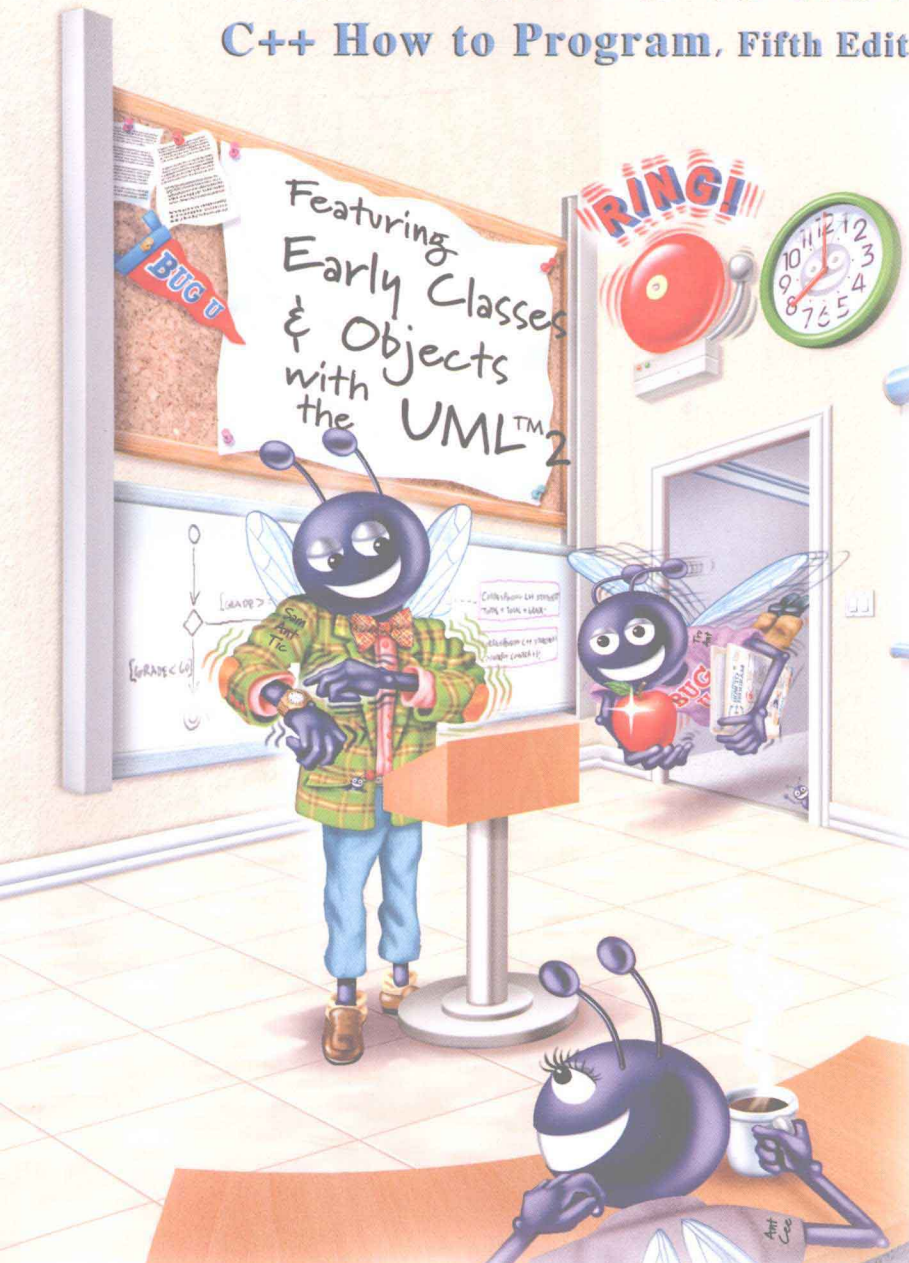
CD includes the book's hundreds of C++ LIVE-CODE examples, links to free C++ compilers and development tools, and hundreds of Web resources, including general references, tutorials, FAQs, newsgroups and STL information.

C++大学教程

——(第五版)(英文版)——

C++ How to Program, Fifth Edition

PEARSON Prentice Hall



EARLY CLASSES/OBJECTS/OOP
- Classes, Objects, Encapsulation
- Inheritance, Polymorphism
- Integrated OOP Case Studies: Time, GradeBook, Employee

FUNDAMENTALS
- History, Hardware, Software
- Stream I/O, Types, Operators
- Control Statements, Functions
- Arrays, Vectors
- Pointers, References
- String Class, C-Style Strings
- Operator Overloading
- Exceptions, Files
- Web Programming
- Bit and Character Manipulation
- GNU C++/Visual C++® Debuggers

DATA STRUCTURES
- Recursion, Searching, Sorting
- Lists, Queues, Stacks, Trees
- Templates
- Standard Template Library: Containers, Iterators, and Algorithms

OOD/UML 2 ATM CASE STUDY (OPTIONAL)
- Determining Classes, Attributes, States, Activities, Operations, Collaborations
- Diagrams: Use Case, Class, State, Activity, Communication, Sequence

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

DEITEL

[美] H. M. Deitel
P. J. Deitel 著

国外计算机科学教材系列

C++ 大学教程

（第五版）

（英文版）

C++ How To Program

Fifth Edition

[美] H. M. Deitel
Deitel & Associates, Inc.
P. J. Deitel
Deitel & Associates, Inc.
著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是一本C++编程方面的优秀教程，全面介绍了过程式编程与面向对象编程的原理与方法，细致地分析了各种性能问题、可移植性问题及可能出错的情况。与第四版相比，本书从内容和结构上都做了较大幅度的修订与有意义的提升，特别是在“尽早接触类和对象”这一观点的指导下，从第1章就开始介绍类和对象的内容，突破了传统的教学模式，使学生直接“考虑对象”和深入掌握面向对象的基本概念。作者通过大量的示例程序，重点突出了利用UML进行面向对象的设计，引入了使用CGI的Web应用开发，并且帮助学生创建真实世界的C++应用程序。本书无论从广度还是深度上来说都非常全面，并且从基础概念讲起，同样适合没有编程经验的读者学习。

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，也是软件设计人员进行C++程序开发的宝贵参考资料。

版权贸易合同登记号：图字：01-2007-5189

图书在版编目（CIP）数据

C++大学教程 = C++ How To Program: 第5版：英文/（美）戴特尔（Deitel, H. M.）著.
北京：电子工业出版社，2008.11
（国外计算机科学教材系列）
ISBN 978-7-121-05579-9

Ⅰ.C.... Ⅱ.①戴... Ⅲ.C语言-程序设计-高等学校-教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第192566号

责任编辑：冯小贝
印　　刷：北京市天竺颖华印刷厂
装　　订：三河市鑫金马印装有限公司
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：70.5　　字数：1985千字
印　　次：2008年11月第1次印刷
定　　价：128.00元（含光盘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zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

C++

How to Program

FIFTH EDITION

Deitel® Books, Cyber Classrooms, Complete Tra published by

How To Program Series

Advanced Java™ 2 Platform How to Program
C How to Program, 4/E
C++ How to Program, 5/E
C# How to Program
e-Business and e-Commerce How to Program
Internet and World Wide Web How to Program, 3/E
Java How to Program, 6/E
Small C++ How to Program, 5/E
Small Java™ How to Program, 6/E
Perl How to Program
Python How to Program
Visual C++® .NET How to Program
Visual Basic® 6 How to Program
Visual Basic® .NET How to Program, 2/E
Wireless Internet & Mobile Business How to Program
XML How to Program

Simply Series

Simply C++: An Application-Driven Tutorial Approach
Simply C#: An Application-Driven Tutorial Approach
Simply Java™ Programming: An Application-Driven Tutorial Approach
Simply Visual Basic® .NET: An Application Driven Tutorial Approach (Visual Studio .NET 2002 Edition)
Simply Visual Basic® .NET: An Application Driven Tutorial Approach (Visual Studio .NET 2003 Edition)

For Managers Series

e-Business and e-Commerce for Managers

.NET How to Program Series

C# How to Program
Visual Basic® .NET How to Program, 2/E
Visual C++® .NET How to Program

Visual Studio® Series

C# How to Program
Visual Basic® .NET How to Program, 2/E
Getting Started with Microsoft® Visual C++® 6 with an Introduction to MFC
Visual Basic® 6 How to Program

Also Available

e-books
CourseCompass, WebCT and Blackboard support
Pearson Choices: SafariX

ining Courses and Web-Based Training Courses Prentice Hall

Multimedia Cyber Classroom and Web-Based Training Series

C++ Multimedia Cyber Classroom, 5/E

C# Multimedia Cyber Classroom

e-Business and e-Commerce Multimedia Cyber Classroom

Internet and World Wide Web Multimedia Cyber Classroom, 2/E

Java™ Multimedia Cyber Classroom, 6/E

Perl Multimedia Cyber Classroom

Python Multimedia Cyber Classroom

Visual Basic® 6 Multimedia Cyber Classroom

Visual Basic® .NET Multimedia Cyber Classroom, 2/E

Wireless Internet & Mobile Business Programming Multimedia Cyber Classroom

XML Multimedia Cyber Classroom

The Complete Training Course Series

The Complete C++ Training Course, 4/E

The Complete C# Training Course

The Complete e-Business and e-Commerce Programming Training Course

The Complete Internet and World Wide Web Programming Training Course, 2/E

The Complete Java™ 2 Training Course, 5/E

The Complete Perl Training Course

The Complete Python Training Course

The Complete Visual Basic® 6 Training Course

The Complete Visual Basic® .NET Training Course, 2/E

The Complete Wireless Internet & Mobile Business Programming Training Course

The Complete XML Programming Training Course

To follow the Deitel publishing program, please register at:

`www.deitel.com/newsletter/subscribe.html`

for the free *DEITEL® BUZZ ONLINE* e-mail newsletter.

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 or`
`www.InformIT.com/deitel`

Library of Congress Cataloging-in-Publication Data
On file

Vice President and Editorial Director, ECS: ***Marcia J. Horton***
Senior Acquisitions Editor: ***Kate Hargett***
Associate Editor: ***Jennifer Cappello***
Assistant Editor: ***Sarah Parker***
Editorial Assistant: ***Michael Giacobbe***
Executive Managing Editor: ***Vince O'Brien***
Managing Editor: ***Tom Manshreck***
Production Editor: ***John F. Lovell***
Production Editor, Media: ***Bob Engelhardt***
Production Assistant: ***Asha Rohra***
Director of Creative Services: ***Paul Belfanti***
A/V Production Editor: ***Xiaohong Zhu***
Art Studio: ***Artworks, York, PA***
Art Director: ***Geoffrey Cassar***
Cover Design: ***Abbey S. Deitel, Harvey M. Deitel, Francesco Santalucia, Geoffrey Cassar***
Interior Design: ***Harvey M. Deitel, Geoffrey Cassar***
Assistant Manufacturing Manager: ***Michael Bell***
Manufacturing Buyer: ***Lisa McDowell***
Marketing Manager: ***Pamela Hersperger***
Marketing Assistant: ***Barrie Reinhold***

Upper Saddle River, New Jersey 07458

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-185757-6

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educacion de Mexico, S.A. de C.V.
Pearson Education–Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, Inc., *Upper Saddle River, New Jersey*

C++ How to Program

FIFTH EDITION

H. M. Deitel

Deitel & Associates, Inc.

P. J. Deitel

Deitel & Associates, Inc.

PEARSON Prentice Hall

Upper Saddle River, New Jersey 07458

Trademarks

Borland and C++ Builder are trademarks or registered trademarks of Borland.

Cygwin is a trademark and copyrighted work of Red Hat, Inc. in the United States and other countries.

Dive Into is a registered trademark of Deitel & Associates, Inc.

GNU is a trademark of the Free Software Foundation.

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.

Linux is a registered trademark of Linus Torvalds.

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

Netscape browser window © 2004 Netscape Communications Corporation. Used with permission. Netscape Communications has not authorized, sponsored, endorsed, or approved this publication and is not responsible for its content.

Object Management Group, OMG, Unified Modeling Language and UML are trademarks of Object Management Group, Inc.

To:

Stephen Clamage

Chairman of the J16 committee, "Programming Language C++" that is responsible for the C++ standard; Senior Staff Engineer, Sun Microsystems, Inc., Software Division.

Don Kostuch

Independent Consultant

and Mike Miller

Former Vice Chairman and Core Language Working Group Chairman of the J16 committee, "Programming Language C++;" Software Design Engineer, Edison Design Group, Inc.

For your mentorship, friendship, and tireless devotion to insisting that we "get it right" and helping us do so.

It is a privilege to work with such consummate C++ professionals.

Harvey M. Deitel and Paul J. Deitel

出 版 说 明

21世纪初的5至10年是我国国民经济和社会发展的重要时期，也是信息产业快速发展的关键时期。在我国加入WTO后的今天，培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡，是我国面对国际竞争时成败的关键因素。

当前，正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期，为使我国教育体制与国际化接轨，有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材，以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验，翻译出版了“国外计算机科学教材系列”丛书，这套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。这些教材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。同时，我们也适当引进了一些优秀英文原版教材，本着翻译版本和英文原版并重的原则，对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上，我们大都选择国外著名出版公司出版的高校教材，如Pearson Education培生教育出版集团、麦格劳－希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者，如道格拉斯·科默（Douglas E. Comer）、威廉·斯托林斯（William Stallings）、哈维·戴特尔（Harvey M. Deitel）、尤利斯·布莱克（Uyless Black）等。

为确保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系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士，也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、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，包括对所选教材进行全面论证；选择编辑时力求达到专业对口；对排版、印制质量进行严格把关。对于英文教材中出现的错误，我们通过与作者联络和网上下载勘误表等方式，逐一进行了修订。

此外，我们还将与国外著名出版公司合作，提供一些教材的教学支持资料，希望能为授课老师提供帮助。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。

电子工业出版社

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教授 中国计算机学会副理事长，数据库专业委员会主任
	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、博士生导师 清华大学深圳研究生院信息学部主任
	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 教育部计算机基础课程教学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

Preface

"The chief merit of language is clearness ..."
—Galen

Welcome to C++ and *C++ How to Program, Fifth Edition*! C++ is a world-class programming language for developing industrial-strength, high-performance computer applications. We believe that this book and its support materials have everything instructors and students need for an informative, interesting, challenging and entertaining C++ educational experience. In this Preface, we overview the many new features of *C++ How to Program, 5/e*. The *Tour of the Book* section of the Preface gives instructors, students and professionals a sense of *C++ How to Program, 5/e*'s coverage of C++ and object-oriented programming. We also overview various conventions used in the book, such as syntax coloring the code examples, "code washing" and code highlighting. We provide information about free compilers that you can find on the Web. We also discuss the comprehensive suite of educational materials that help instructors maximize their students' learning experience, including the *Instructor's Resource CD*, PowerPoint® Slide lecture notes, course management systems, SafariX (Pearson Education's WebBook publications) and more.

Features of C++ How to Program, 5/e

At Deitel & Associates, we write college-level computer science textbooks and professional books. To create *C++ How to Program, 5/e*, we put the previous edition of *C++ How to Program* under the microscope. The new edition has many compelling features:

- **Major Content Revisions**. All the chapters have been significantly updated and upgraded. We tuned the writing for clarity and precision. We also adjusted our use of C++ terminology in accordance with the ANSI/ISO C++ standard document that defines the language.
- **Smaller Chapters**. Larger chapters have been divided into smaller, more manageable chapters (e.g., Chapter 1 of the Fourth Edition has been split into Chapters 1-2; Chapter 2 of the Fourth Edition is now Chapters 4-5).
- **Early Classes and Objects Approach**. We changed to an early classes and objects pedagogy. Students are introduced to the basic concepts and terminology of object technology in Chapter 1. In the previous edition, students began developing customized, reusable classes and objects in Chapter 6, but in this edition, they do so in our completely new Chapter 3. Chapters 4-7 have been carefully rewritten from an "early classes and objects" perspective. This new edition is object oriented, where appropriate, from the start and throughout the text. Moving the discussion of objects and classes to earlier chapters gets students "thinking about objects" immediately and mastering these concepts more completely. Object-ori-ented programming is not trivial by any means, but it's fun to write object-oriented programs, and students can see immediate results.
- **Integrated Case Studies**. We have added several case studies spanning multiple sections and chapters that often build on a class introduced earlier in the book to demonstrate new programming concepts later in the book. These case studies include the development of the `GradeBook` class in Chapters 3-7, the `Time` class in several sections of Chapters 9-10, the `Employee` class in Chapters 12-13, and the optional OOD/UML ATM case study in chapters 1-7, 9, 13 and Appendix G.
- **Integrated `GradeBook` Case Study**. We added a new `GradeBook` case study to reinforce our early classes presentation. It uses classes and objects in Chapters 3-7 to incrementally build a `GradeBook` class that represents an instructor's grade book and performs various calculations based on a set of student grades, such as calculating the average grade, finding the maximum and minimum and printing a bar chart.
- **Unified Modeling Language™ 2.0 (UML 2.0) — Introducing the UML 2.0**. The Unified Modeling Language (UML) has become the preferred graphical modeling language for designers of object-oriented systems. All the

UML diagrams in the book comply with the new UML 2.0 specification. We use UML class diagrams to visually represent classes and their inheritance relationships, and we use UML activity diagrams to demonstrate the flow of control in each of C++'s control statements. We make especially heavy use of the UML in the optional OOD/UML ATM case study

- **Optional OOD/UML ATM Case Study**. We replaced the optional elevator simulator case study from the previous edition with a new optional OOD/UML automated teller machine (ATM) case study in the Software Engineering Case Study sections of Chapters 1-7, 9 and 13. The new case study is simpler, smaller, more "real world" and more appropriate for first and second programming courses. The nine case study sections present a carefully paced introduction to objectoriented design using the UML. We introduce a concise, simplified subset of the UML 2.0, then guide the reader through a first design experience intended for the novice object-oriented designer/programmer. Our goal in this case study is to help students develop an object-oriented design to complement the object-oriented programming concepts they begin learning in Chapter 1 and implementing in Chapter 3. The case study was reviewed by a distinguished team of OOD/UML academic and industry professionals. The case study is not an exercise; rather, it is a fully developed end-to-end learning experience that concludes with a detailed walkthrough of the complete 877-line C++ code implementation. We take a detailed tour of the nine sections of this case study later in the Preface.
- **Compilation and Linking Process for Multiple-Source-File Programs**. Chapter 3 includes a detailed diagram and discussion of the compilation and linking process that produces an executable application.
- **Function Call Stack Explanation**. In Chapter 6, we provide a detailed discussion (with illustrations) of the function call stack and activation records to explain how C++ is able to keep track of which function is currently executing, how automatic variables of functions are maintained in memory and how a function knows where to return after it completes execution.
- **Early Introduction of C++ Standard Library string and vector Objects**. The `string` and `vector` classes are used to make earlier examples more object-oriented.
- **Class `string`**. We use class `string` instead of C-like pointer-based `char *` strings for most string manipulations throughout the book. We continue to include discussions of `char *` strings in Chapter 8, 10, 11 and 22 to give students practice with pointer manipulations, to illustrate dynamic memory allocation with new and `delete`, to build our own `String` class, and to prepare students for assignments in industry where they will work with `char *` strings in C and C++ legacy code.
- **Class Template `vector`**. We use class template `vector` instead of C-like pointerbased array manipulations throughout the book. However, we begin by discussing C-like pointer-based arrays in Chapter 7 to prepare students for working with C and C++ legacy code in industry and to use as a basis for building our own customized `Array` class in Chapter 11, Operating Overloading.
- **Tuned Treatment of Inheritance and Polymorphism**. Chapters 12-13 have been carefully tuned, making the treatment of inheritance and polymorphism clearer and more accessible for students who are new to OOP. An `Employee` hierarchy replaces the `Point/Circle/Cylinder` hierarchy used in prior editions to introduce inheritance and polymorphism. The new hierarchy is more natural.
- **Discussion and Illustration of How Polymorphism Works "Under the Hood."** Chapter 13 contains a detailed diagram and explanation of how C++ can implement polymorphism, `virtual` functions and dynamic binding internally. This gives students a solid understanding of how these capabilities really work. More importantly, it helps students appreciate the overhead of polymorphism — in terms of additional memory consumption and processor time. This helps students determine when to use polymorphism and when to avoid it.
- **Web Programming**. Chapter 19, Web Programming, has everything readers need to begin developing their own Web-based applications that will run on the Internet! Students will learn how to build so-called *n*-tier applications, in which the functionality provided by each tier can be distributed to separate computers across the Internet or executed on the same computer. Using the popular Apache HTTP server (which is available free for download from `www.apache.org`) we present the CGI (common Gateway Interface) protocol and discuss how CGI allows a Web server to communicate with the top tier (e.g., a Web browser running on the user's computer) and CGI

scripts (i.e., our C++ programs) executing on a remote system. The chapter examples conclude with an e-business case study of an online bookstore that allows users to add books to an electronic shopping cart.

- **Standard Template Library (STL).** This might be one of the most important topic in the book in terms of your appreciation of software reuse. The STL defines powerful, template-based, reusable components that implement many common data structures and algorithms used to process those data structures. Chapter 23 introduces the STL and discusses its three key components — containers, iterators and algorithms. We show that using STL components provides tremendous expressive power and can reduce many lines of code to a single statement.
- **XHTML.** The World Wide Web Consortium (W3C) has declared HyperText Markup Language (HTML) to be a legacy technology that will undergo no further development. HTML is being replaced by the Extensible HyperText Markup Language (XHTML) — an XML-based technology that rapidly is becoming the standard for describing Web content. We use XHTML in Chapter 19, Web Programming; Appendix J and Appendix K introduce XHTML.
- **ANSI/ISO C++ Standard Compliance.** We have audited our presentation against the most recent ANSI/ISO C++ standard document for completeness and accuracy. [*Note:* If you need additional technical details on C++, you may want to read the C++ standard document. An electronic PDF copy of the C++ standard document, number INCITS/ISO/IEC 14882-2003, is available for $18 at `webstore.ansi.org/ansidocstore/default.asp`.]
- **New Debugger Appendices.** We include two new Using the Debugger appendices — Appendix L, Using the Visual Studio .NET Debugger, and Appendix M, Using the GNU C++ Debugger.
- **New Interior Design.** Working with the creative services team at Prentice Hall, we redesigned the interior styles for our *How to Program* Series. The new fonts are easier on the eyes and the new art package is more appropriate for the more detailed illustrations. We now place the defining occurrence of each key term both in the text and in the index in blue, bold style text for easier reference. We emphasize on-screen components in the bold **Helvetica** font (e.g., the **File** menu) and emphasize C++ program text in the `Lucida` font (e.g., `int x = 5;`).
- **Syntax Coloring.** We syntax color all the C++ code, which is consistent with most C++ integrated development environments and code editors. This greatly improves code readability — an especially important goal, given that this book contains 17,292 lines of code. Our syntax-coloring conventions are as follows[1]:

```
comments appear in green
keywords appear in dark blue
constants and literal values appear in light blue
errors appear in red
all other code appears in black
```

- **Code Highlighting.** Extensive code highlighting makes it easy for readers to locate each program's new features and helps students review the material rapidly when preparing for exams or labs.
- **"Code washing."** This is our term for using extensive and meaningful comments, using meaningful identifiers, applying uniform indentation conventions, aligning curly braces vertically, using a `// end...` comment on every line with a right curly brace and using vertical spacing to highlight significant program units such as control statements and functions. This process results in programs that are easy to read and self-documenting. We have extensively "code washed" all of the source-code programs in both the text and the book's ancillaries. We have worked hard to make our code exemplary.
- **Code Testing on Multiple Platforms.** We tested the code examples on various popular C++ platforms. For the most part, all of the book's examples port easily to all popular ANSI/ISO standard-compliant compilers. We will post any problems at `www.deitel.com/books/cpphtp5/index.html`.
- **Errors and Warnings Shown for Multiple Platforms.** For programs that intentionally contain errors to illustrate a key concept, we show the error messages that result on several popular platforms.
- **Large Review Team.** The book has been carefully scrutinized by a team of 30 distinguished academic and industry reviewers (listed later in the Preface).
- **Free Web-Based *Cyber Classroom*.** We've converted our popular interactive multimedia version of the text (which we call a *Cyber Classroom*) from a for-sale, CDbased product to a free online supplement, available with new books purchased from Prentice Hall for fall 2005 classes.

1 由于印刷原因，本书代码段中已取消颜色区分。

- **Free Student Solutions Manual.** We've converted our Student Solutions Manual, which contains solutions to approximately half of the exercises, from a for-sale softcover book to a free online supplement, available with new books purchased from Prentice Hall for fall 2005 classes.
- **Free Lab Manual.** We've converted our Lab Manual, *C++ in the Lab*, from a forsale softcover book to a free online supplement included with the *Cyber Classroom*, available with new books purchased from Prentice Hall for fall 2005 classes.

As you read this book, if you have questions, send an e-mail to `deitel@deitel.com`; we will respond promptly. Please visit our Web site, `www.deitel.com` and be sure to sign up for the free *D EITEL® Buzz Online* e-mail newsletter at `www.deitel.com/newsletter/subscribe.html` for updates to this book and the latest information on C++. We also use the Web site and the newsletter to keep our readers and industry clients informed of the latest news on Deitel publications and services. Please check the following Web site regularly for errata, updates regarding the C++ software, free downloads and other resources:

`www.deitel.com/books/cpphtp5/index.html`

Teaching Approach

C++ How to Program, 5/e contains an abundant collection of examples, exercises and projects drawn from many fields to provide the student with a chance to solve interesting real-world problems. The book concentrates on the principles of good software engineering and stresses program clarity. We avoid arcane terminology and syntax specifications in favor of teaching by example. We are educators who teach programming languages courses in industry classrooms worldwide. Dr. Harvey M. Deitel has 20 years of college teaching experience, including serving as chairman of the Computer Science Department at Boston College, and 15 years of industry teaching experience. Paul Deitel has 12 years of industry teaching experience. The Deitels have taught C++ courses at all levels to the government, industry, military and academic clients of Deitel & Associates.

Learning C++ using the LIVE-CODE Approach

C++ How to Program, 5/e, is loaded with C++ programs—each new concept is presented in the context of a complete working C++ program that is immediately followed by one or more sample executions showing the program's inputs and outputs. This style exemplifies the way we teach and write about programming. We call this method of teaching and writing the L IVE-CODE Approach. *We use programming languages to teach programming languages.* Reading the examples in the text is much like typing and running them on a computer. We provide all the source code for the book's examples on the accompanying CD and at `www.deitel.com`—making it easy for students to run each example as they study it.

World Wide Web Access

All the source-code examples for *C++ How to Program, 5/e* (and our other publications) are available on the Internet as downloads from

`www.deitel.com`

Registration is quick and easy, and the downloads are free. We suggest downloading all the examples (or copying them from the CD included in the back of this book), then running each program as you read the corresponding text. Making changes to the examples and immediately seeing the effects of those changes is a great way to enhance your C++ learning experience.

Objectives

Each chapter begins with a statement of objectives. This lets students know what to expect and gives them an opportunity, after reading the chapter, to determine if they have met these objectives. This is a confidence builder and a source of positive reinforcement.

Quotations

The learning objectives are followed by quotations. Some are humorous, some philosophical and some offer interesting insights. We hope that you will enjoy relating the quotations to the chapter material. Many of the quotations are worth a second look after reading the chapter.

Outline

The chapter outline helps students approach the material in a top-down fashion, so they can anticipate what is to come, and set a comfortable and effective learning pace.

17,292 Lines of Syntax-Colored Code in 260 Example Programs with Program Inputs and Outputs)

Our LIVE-CODE programs range in size from just a few lines of code to more substantial examples. Each program is followed by a window containing the input/output dialogue produced when the program is run, so students can confirm that the programs run as expected. Relating outputs to the program statements that produce them is an excellent way to learn and to reinforce concepts. Our programs demonstrate the diverse features of C++. The code is line-numbered and syntax colored—with C++ keywords, comments and other program text each appearing in different colors. This facilitates reading the code — students will especially appreciate the syntax coloring when they read the larger programs.

735 Illustrations/Figures

An abundance of charts, tables, line drawings, programs and program outputs is included. We model the flow of control in control statements with UML activity diagrams. UML class diagrams model the data members, constructors and member functions of classes. We use additional types of UML diagrams throughout our optional OOD/UML ATM Software Engineering Case Study.

571 Programming Tips

We include programming tips to help students focus on important aspects of program development. We highlight these tips in the form of *Good Programming Practices*, *Common Programming Errors*, *Performance Tips*, *Portability Tips*, *Software Engineering Observations* and *Error-Prevention Tips*. These tips and practices represent the best we have gleaned from a combined six decades of programming and teaching experience. One of our students, a mathematics major, told us that she feels this approach is like the highlighting of axioms, theorems, lemmas and corollaries in mathematics books — it provides a basis on which to build good software.

Good Programming Practices

Good Programming Practices *are tips for writing clear programs. These techniques help students produce programs that are more readable, self-documenting and easier to maintain.*

Common Programming Errors

Students who are new to programming (or a programming language) tend to make certain errors frequently. Focusing on these Common Programming Errors *reduces the likelihood that students will make the same mistakes and shortens long lines outside instructors' offices during office hours!*

Performance Tips

In our experience, teaching students to write clear and understandable programs is by far the most important goal for a first programming course. But students want to write the programs that run the fastest, use the least memory, require the smallest number of keystrokes or dazzle in other nifty ways. Students really care about performance. They want to know what they can do to "turbo charge" their programs. So we highlight opportunities for improving program performance — making programs run faster or minimizing the amount of memory that they occupy.

Portability Tips

Software development is a complex and expensive activity. Organizations that develop software must often produce versions customized to a variety of computers and operating systems. So there is a strong emphasis today on portability, i.e., on producing software that will run on a variety of computer systems with few, if any, changes. Some programmers assume that if they implement an application in standard C++, the application will be portable. This simply is not the case. Achieving portability requires careful and cautious design. There are many pitfalls. We include Portability Tips to help students write portable code and to provide insights on how C++ achieves its high degree of portability.

Software Engineering Observations

The object-oriented programming paradigm necessitates a complete rethinking of the way we build software systems. C++ is an effective language for achieving good software engineering. The Software Engineering Observations *highlight architectural and design issues that affect the construction of software systems, especially large-scale systems. Much of what the student learns here will be useful in upper-level courses and in industry as the student begins to work with large, complex real-world systems.*