Pearson

国外计算机科学教材系列

C++大学教程

(第九版)(英文版)

[美] Paul Deitel
Harvey Deitel 著

C++ How to Program

Ninth Edition

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

需要全本请访问 www.ertongbook.com

国外计算机科学教材系列

C++大学教程

（第九版）

（英文版）

C++ How to Program

Ninth Edition

[美] Paul Deitel
Harvey Deitel 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++编程方面的优秀教材，全面介绍了面向对象编程的原理与方法，详细分析了与 C++编程有关的技术。本书的主要内容包括类与对象、控制语句、函数与递归、数组、指针、运算符重载、继承、多态、输入/输出流、异常处理、文件处理、模板、搜索与排序等。全书以“活代码”的方式详细分析了每个知识要点，提供了丰富的自测练习和项目练习，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 C++语言课程的双语教材，也是软件设计人员学习 C++编程的理想读物。

Authorized reprint from the English language edition , entitled C++ How to Program, International Edition, Ninth Edition, ISBN 9780273793298 by Paul Deitel and Harvey Deitel, published by Pearson Education, Ltd., Copyright © Pearson Education Limited 2014.

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.

English language edition published by PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright © 2019.

This edition is manufactured in the People's Republic of China, and is authorized for sale and distribution only in the mainland 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：01-2018-6063

图书在版编目(CIP)数据

C++大学教程：第九版：英文 /(美)保罗·戴特尔(Paul Deitel)，(美)哈维·戴特尔(Harvey Deitel)著. — 北京：电子工业出版社，2019.3

(国外计算机科学教材系列)

书名原文：C++ How to Program，Ninth Edition

ISBN 978-7-121-29542-3

I. ①C… II. ①保… ②哈… III. ①C 语言—程序设计—高等学校—教材—英文 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 030269 号

策划编辑：冯小贝
责任编辑：冯小贝
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
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：53.75　字数：2035 千字
版　　次：2005 年 9 月第 1 版(原著第 4 版)
　　2019 年 3 月第 5 版(原著第 9 版)
印　　次：2019 年 3 月第 1 次印刷
定　　价：159.00 元

凡所购买电子工业出版社图书有缺损问题，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)88254888，88258888。

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。

本书咨询联系方式：fengxiaobei@phei.com.cn。

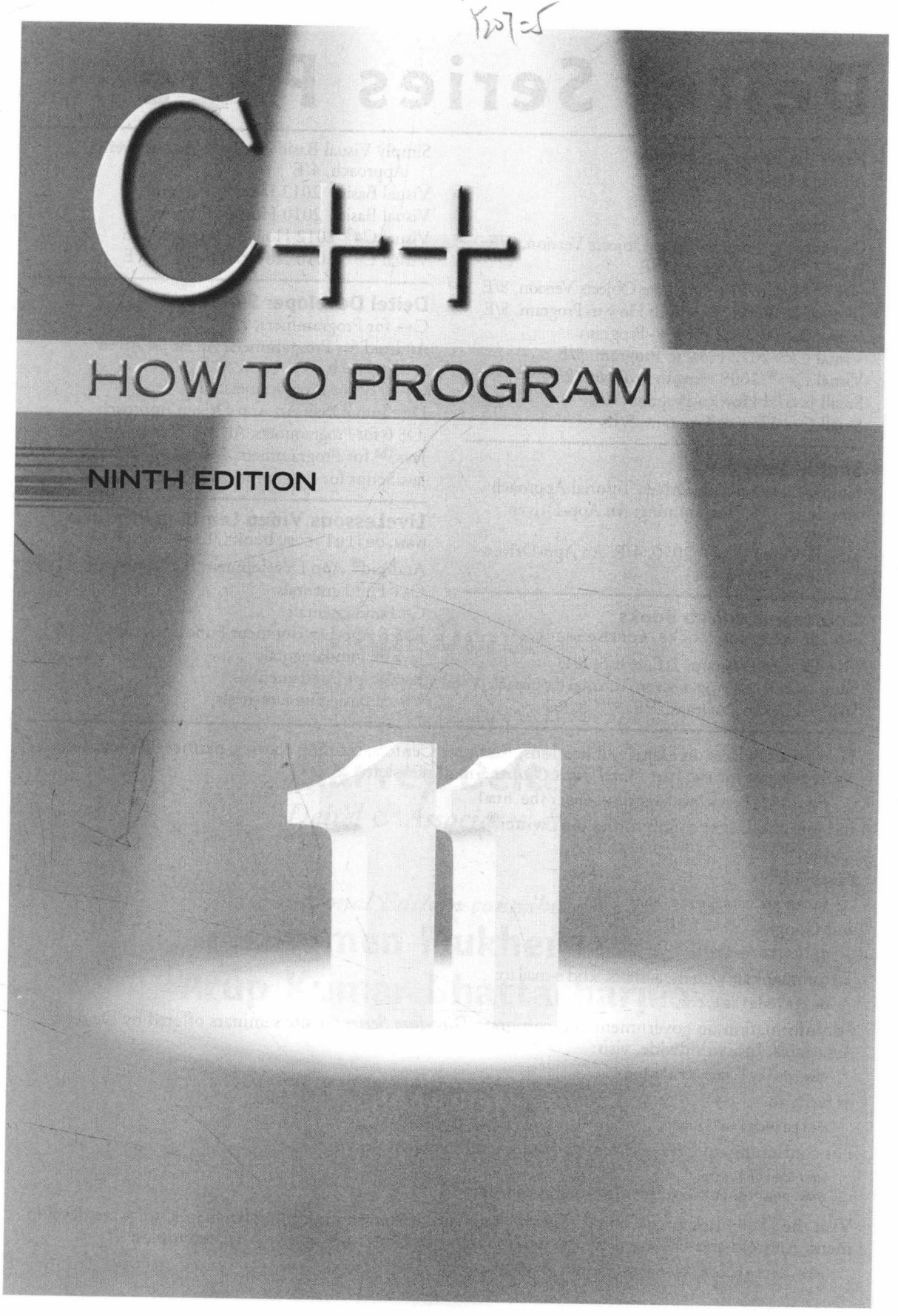
TP312
Y207=5

C++

HOW TO PROGRAM

NINTH EDITION

11



Deitel Series Page

How To Program Series

Android How to Program
C How to Program, 7/E
C++ How to Program, 9/E
C++ How to Program, Late Objects Version, 7/E
Java™ How to Program, 9/E
Java™ How to Program, Late Objects Version, 8/E
Internet & World Wide Web How to Program, 5/E
Visual Basic® 2012 How to Program
Visual C#® 2012 How to Program, 3/E
Visual C++® 2008 How to Program, 2/E
Small Java™ How to Program, 6/E
Small C++ How to Program, 5/E

Simply Series

Simply C++: An App-Driven Tutorial Approach
Simply Java™ Programming: An App-Driven Tutorial Approach
Simply Visual Basic® 2010, 4/E: An App-Driven Tutorial Approach

CourseSmart Web Books

www.deitel.com/books/CourseSmart/

C++ How to Program, 7/E, 8/E & 8/E
Simply C++: An App-Driven Tutorial Approach
Java™ How to Program, 7/E, 8/E & 9/E
Simply Visual Basic 2010: An App-Driven Approach, 4/E
Visual Basic® 2012 How to Program
Visual Basic® 2010 How to Program
Visual C#® 2012 How to Program, 5/E
Visual C#® 2010 How to Program, 4/E

Deitel Developer Series

C++ for Programmers, 2/E
Android for Programmers: An App-Driven Approach
C# 2010 for Programmers, 3/E
Dive Into iOS 6: An App-Driven Approach
iOS 6 for Programmers: An App-Driven Approach
Java™ for Programmers, 2/E
JavaScript for Programmers

LiveLessons Video Learning Products

www.deitel.com/books/LiveLessons/

Android® App Development Fundamentals
C++ Fundamentals
C# Fundamentals
iOS 6 App Development Fundamentals
Java™ Fundamentals
JavaScript Fundamentals
Visual Basic Fundamentals

To receive updates on Deitel publications, Resource Centers, training courses, partner offers and more, please register for the free *Deitel Buzz Online* e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

and join the Deitel communities on Twitter®

@deitel

Facebook®

facebook.com/DeitelFan

and Google+

gplus.to/deitel

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

deitel@deitel.com

For information on government and corporate *Dive-Into Series* on-site seminars offered by Deitel & Associates, Inc. worldwide, visit:

www.deitel.com/training/

or write to

deitel@deitel.com

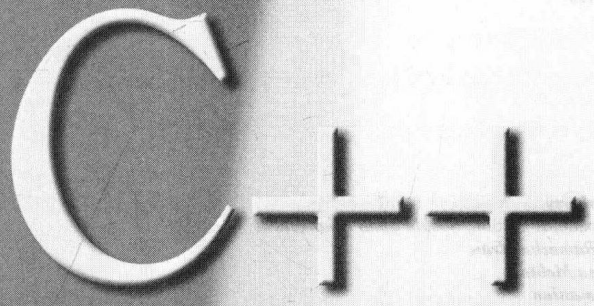
For continuing updates on Prentice Hall/Deitel publications visit:

www.deitel.com
www.pearsoninternationaleditions.com/deitel

Visit the Deitel Resource Centers that will help you master programming languages, software development, Android and iPhone/iPad app development, and Internet- and web-related topics:

www.deitel.com/ResourceCenters.html

需要全本请访问 www.ertongbook.com





HOW TO PROGRAM

NINTH EDITION

Paul Deitel
Deitel & Associates, Inc.

Harvey Deitel
Deitel & Associates, Inc.

International Edition contributions by
Soumen Mukherjee
Arup Kumar Bhattacharjee

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montréal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Vice President and Editorial Director: ***Marcia J. Horton***
Executive Editor: ***Tracy Johnson***
Associate Editor: ***Carole Snyder***
Director of Marketing: ***Christy Lesko***
Marketing Manager: ***Yezan Alayan***
Marketing Assistant: ***Jon Bryant***
Director of Production: ***Erin Gregg***
Managing Editor: ***Scott Disanno***
Associate Managing Editor: ***Robert Engelhardt***
Publisher, International Edition: ***Angshuman Chakraborty***
Publishing Administrator and Business Analyst, International Edition: ***Shokhi Shah Khandelwal***
Acquisitions Editor, International Edition: ***Shivangi Ramachandran***
Publishing Administrator, International Edition: ***Hema Mehta***
Project Editor, International Edition: ***Karthik Subramanian***
Senior Manufacturing Controller, Production, International Edition: ***Trudy Kimber***
Operations Specialist: ***Lisa McDowell***
Art Director: ***Anthony Gemmellaro***
Cover Design: ***Jodi Notowitz***
Cover Photo Credit: ***lebbid/Shutterstock***
Media Project Manager: ***Renata Butera***

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsoninternationaleditions.com

© Pearson Education Limited 2014

The rights of Paul Deitel and Harvey Deitel to be identified as authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

Authorized adaptation from the United States edition, entitled C++ How to Program, Ninth Edition, ISBN 978-0-13-337871-9, *by Paul Deitel and Harvey Deitel, published by Pearson Education © 2014.*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6-10 Kirby Street, London EC1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

ISBN 10: 0-273-79329-2
ISBN 13: 978-0-273-79329-8

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1
14 13 12 11 10

Typeset in AGaramond Regular by GEX Publishing Services

Printed and bound by Courier Westford in The United States of America
The publisher's policy is to use paper manufactured from sustainable forests.

In memory of Dennis Ritchie,
creator of the C programming language—
one of the key languages that inspired C++.

Paul and Harvey Deitel

Trademarks

Carnegie Mellon Software Engineering Institute is a trademark of Carnegie Mellon University.

CERT® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screen shots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

UNIX is a registered trademark of The Open Group.

Throughout this book, trademarks are used. Rather than put a trademark symbol in every occurrence of a trademarked name, we state that we are using the names in an editorial fashion only and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Preface

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

Welcome to the C++ computer programming language and *C++ How to Program, Ninth Edition*. This book presents leading-edge computing technologies. It's appropriate for introductory course sequences based on the curriculum recommendations of two key professional organizations—the ACM and the IEEE. If you haven't already done so, please read the back cover and inside back cover—these capture the essence of the book concisely. In this Preface we provide more detail for students, instructors and professionals.

At the heart of the book is the Deitel signature *live-code approach*—we present concepts in the context of complete working programs followed by sample executions, rather than in code snippets. Read the online Before You Begin section(`www.deitel.com/books/cpphtp9/cpphtp9_BYB.pdf`) to learn how to set up your Linux-based, Windows-based or Apple OS X-based computer to run the hundreds of code examples. All the source code is available at `www.deitel.com/books/cpphtp9` and `www.pearsoninternational-editions.com/deitel`. Use the source code we provide to *run each program* as you study it.

We believe that this book and its support materials will give you an informative, challenging and entertaining introduction to C++. As you read the book, if you have questions, we're easy to reach at `deitel@deitel.com`—we'll respond promptly. For book updates, visit `www.deitel.com/books/cpphtp9`, join our social media communities on Facebook(`www.deitel.com/DeitelFan`), Twitter(`@deitel`), Google+ (`gplus.to/deitel`) and LinkedIn(`bit.ly/DeitelLinkedIn`), and subscribe to the *Deitel Buzz® Online* newsletter (`www.deitel.com/newsletter/subscribe.html`).

C++11 Standard

The new C++11 standard, published in 2011, motivated us to write *C++ How to Program, 9/e*. Throughout the book, each new C++11 feature is marked with the "11" icon you see here in the margin. These are some of the key C++11 features of this new edition:

- ***Conforms to the new C++11 standard.*** Extensive coverage of the new C++11 features (Fig. 1).
- ***Code thoroughly tested on three popular industrial-strength C++11 compilers.*** We tested the code examples on GNU C++ 4.7, Microsoft® Visual C++® 2012 and Apple® LLVM in Xcode® 4.5.
- ***Smart pointers.*** Smart pointers help you avoid dynamic memory management errors by providing additional functionality beyond that of built-in pointers. We discuss `unique_ptr` in Chapter 17, and `shared_ptr` and `weak_ptr` in Chapter 24.
- ***Earlier coverage of Standard Library containers, iterators and algorithms, enhanced with C++11 capabilities.*** We moved the treatment of Standard Library containers, iterators and algorithms from Chapter 22 in the previous edition to Chapters 15 and 16 and enhanced it with additional C++11 features. The vast majority of your data structure needs can be fulfilled by *reusing* these Standard Library capabilities. We'll show you how to build your own *custom* data structures in Chapter 19.

C++11 features in *C++ How to Program, 9/e*		
`all_of` algorithm	Inheriting base-class constructors	Non-deterministic random number generation
`any_of` algorithm	`insert` container member functions return iterators	`none_of` algorithm
`array` container	`is_heap` algorithm	Numeric conversion functions
`auto` for type inference	`is_heap_until` algorithm	`nullptr`
`begin`/`end` functions	Keywords new in C++11	`override` keyword
`cbegin`/`cend` container member functions	Lambda expressions	Range-based `for` statement
Compiler fix for >> in template types	List initialization of key–value `pairs`	Regular expressions
`copy_if` algorithm	List initialization of `pair` objects	Rvalue references
`copy_n` algorithm	List initialization of return values	Scoped `enum`s
`crbegin`/`crend` container member functions	List initializing a dynamically allocated array	`shared_ptr` smart pointer
`decltype`	List initializing a `vector`	`shrink_to_fit` `vector`/`deque` member function
Default type arguments in function templates	List initializers in constructor calls	Specifying the type of an enum's constants
`defaulted` member functions	`long long int` type	`static_assert` objects for file names
Delegating constructors	`min` and `max` algorithms with `initializer_list` parameters	`string` objects for file names
`deleted` member functions	`minmax` algorithm	`swap` non-member function
`explicit` conversion operators	`minmax_element` algorithm	Trailing return types for functions
`final` classes	`move` algorithm	`tuple` variadic template
`final` member functions	Move assignment operators	`unique_ptr` smart pointer
`find_if_not` algorithm	`move_backward` algorithm	Unsigned `long long int`
`forward_list` container	Move constructors	`weak_ptr` smart pointer
Immutable keys in associative containers	`noexcept`	
In-class initializers		

Fig. 1 | A sampling of C++11 features in *C++ How to Program, 9/e*.

- ***Online Chapter 24, C++11: Additional Features.*** In this chapter, we present additional C++11 topics. The new C++11 standard has been available since 2011, but not all C++ compilers have fully implemented the features. If all three of our key compilers already implemented a particular C++11 feature at the time we wrote this book, we generally integrated a discussion of that feature into the text with a live-code example. If any of these compilers had *not* implemented that feature, we included a bold italic heading followed by a brief discussion of the feature. Many of those discussions are expanded in online Chapter 24 as the features are implemented. This chapter includes discussions of regular expressions, `shared_ptr` and `weak_ptr` smart pointers, move semantics and more.
- ***Random Number generation, simulation and game playing.*** To help make programs more secure, we've added a treatment of C++11's new non-deterministic random-number generation capabilities.

Object-Oriented Programming

- ***Early-objects approach.*** The book introduces the basic concepts and terminology of object technology in Chapter 1. You'll develop your first customized classes and objects in Chapter 3. Presenting objects and classes early gets you "thinking about objects" immediately and mastering these concepts more thoroughly.[①]
- ***C++ Standard Library string.*** C++ offers *two* types of strings—`string` class objects (which we begin using in Chapter 3) and C strings. We've replaced most occurrences of C strings with instances of C++

① For courses that require a late-objects approach, consider *C++ How to Program, Late Objects Version*, which begins with six chapters on programming fundamentals (including two on control statements) and continues with seven chapters that gradually introduce object-oriented programming concepts.

需要全本请访问 www.ertongbook.com

class `string` to make programs more robust and eliminate many of the security problems of C strings. We continue to discuss C strings later in the book to prepare you for working with the legacy code that you'll encounter in industry. In new development, you should favor `string` objects.

- ***C++ Standard Library array.*** Our primary treatment of arrays now uses the Standard Library's `array` class template instead of built-in, C-style, pointer-based arrays. We still cover built-in arrays because they remain useful in C++ and so that you'll be able to read legacy code. C++ offers *three* types of arrays—`array`s and `vector`s (which we start using in Chapter 7) and C-style, pointer-based arrays which we discuss in Chapter 8. As appropriate, we use class template `array` instead of C arrays throughout the book. In new development, you should favor class template `array` objects.
- ***Crafting valuable classes.*** A key goal of this book is to prepare you to build valuable classes. In the Chapter 10 case study, you'll build your own custom `Array` class, then in the Chapter 18 exercises you'll convert it to a class template. You'll truly appreciate the class concept. Chapter 10 begins with a test-drive of class template `string` so you can see an elegant use of operator overloading before you implement your own customized class with overloaded operators.
- ***Case studies in object-oriented programming.*** We provide case studies that span multiple sections and chapters and cover the software development lifecycle. These include the `GradeBook` class in Chapters 3–7, the `Time` class in Chapter 9 and the `Employee` class in Chapters 11–12. Chapter 12 contains a detailed diagram and explanation of how C++ can implement polymorphism, `virtual` functions and dynamic binding "under the hood."
- ***Optional case study: Using the UML to develop an object-oriented design and C++ implementation of an ATM.*** The UML(Unified Modeling Language) is the industry-standard graphical language for modeling object-oriented systems. We introduce the UML in the early chapters. Online Chapters 25 and 26 include an *optional* case study on object-oriented design using the UML. We design and implement the software for a simple automated teller machine(ATM). We analyze a typical requirements document that specifies the system to be built. We determine the classes needed to implement that system, the attributes the classes need to have, the behaviors the classes need to exhibit and we specify how the classes must interact with one another to meet the system requirements. From the design we produce a complete C++ implementation. Students often report that the case study helps them "tie it all together" and truly understand object orientation.
- ***Exception handling.*** We integrate basic exception handling *early* in the book. Instructors can easily pull more detailed material forward from Chapter 17, Exception Handling: A Deeper Look.
- ***Custom template-based data structures.*** We provide a rich multi-chapter treatment of data structures—see the Data Structures module in the chapter dependency chart (Fig. 5).
- ***Three programming paradigms.*** We discuss *structured programming*, *object-oriented programming* and *generic programming*.

Pedagogic Features

- ***Rich coverage of C++ fundamentals.*** We include a clear two-chapter treatment of control statements and algorithm development.
- ***Chapter 2 provides a simple introduction to C++ programming.***

- ***Examples.*** We include a broad range of example programs selected from computer science, business, simulation, game playing and other topics (Fig. 2).
- ***Audience.*** The examples are accessible to computer science, information technology, software engineering and business students in novice-level and intermediate-level C++ courses. The book is also used by professional programmers.

Examples	
`Array` class case study	Craps dice game simulation
`Author` class	Credit inquiry program
Bank account program	`Date` class
Bar chart printing program	Downcasting and runtime type information
`BasePlusCommissionEmployee` class	`Employee` class
Binary tree creation and traversal	`explicit` constructor
`BinarySearch` test program	`fibonacci` function
Card shuffling and dealing	`fill` algorithms
`ClientData` class	Function-template specializations of function template `printArray`
`CommissionEmployee` class	`generate` algorithms
Comparing `strings`	`GradeBook` Class
Compilation and linking process	Initializing an array in a declaration
Compound interest calculations with `for`	Input from an `istringstream` object
Converting `string` objects to C strings	Iterative factorial solution
Counter-controlled repetition	`SalesPerson` class
Lambda expressions	Searching and sorting algorithms of the Standard Library
Linked list manipulation	Sequential files
`map` class template	`set` class template
Mathematical algorithms of the Standard Library	`shared_ptr` program
`maximum` function template	`stack` adapter class
Merge sort program	`Stack` class
`multiset` class template	Stack unwinding
`new` throwing `bad_alloc` on failure	Standard Library `string` class program
`PhoneNumber` class	Stream manipulator `showbase`
Poll analysis program	`string` assignment and concatenation
Polymorphism demonstration	`string` member function `substr`
Preincrementing and postincrementing	Summing integers with the `for` statement
`priority_queue` adapter class	`Time` class
`queue` adapter class	`unique_ptr` object managing dynamically allocated memory
Random-access files	Validating user input with regular expressions
Random number generation	`vector` class template
Recursive function `factorial`	
Rolling a six-sided die 6,000,000 times	
`SalariedEmployee` class	

Fig. 2 | A sampling of the book's examples.

- ***Self-Review Exercises and Answers.*** Extensive self-review exercises *and* answers are included for self-study.
- ***Interesting, entertaining and challenging exercises.*** Each chapter concludes with a substantial set of exercises, including simple recall of important terminology and concepts, identifying the errors in code samples, writing individual program statements, writing small portions of C++ classes and member and non-member functions, writing complete programs and implementing major projects. Fig. 3 lists a sampling of the book's exercises, including our *Making a Difference* exercises, which encourage you to use computers and the Internet to research and solve significant social problems. We hope you'll approach these exercises with *your own* values, politics and beliefs.
- ***Illustrations and figures.*** Abundant tables, line drawings, UML diagrams, programs and program outputs are included. A sampling of the book's drawings and diagrams is shown in Fig. 4.

Exercises		
Airline Reservations System	Bubble Sort	Calculating Salaries
Advanced String-Manipulation Exercises	Build Your Own Compiler	CarbonFootprint Abstract Class: Polymorphism
Card Shuffling and Dealing	Build Your Own Computer	Pythagorean Triples
Computer-Assisted Instruction	Eight Queens	Salary Calculator
Computer-Assisted Instruction: Difficulty Levels	Emergency Response	Sieve of Eratosthenes
Computer-Assisted Instruction: Monitoring Student Performance	Enforcing Privacy with Cryptography	Simple Decryption
Computer-Assisted Instruction: Reducing Student Fatigue	Facebook User Base Growth	Simple Encryption
Computer-Assisted Instruction: Varying the Types of Problems	Fibonacci Series	SMS Language
Cooking with Healthier Ingredients	Gas Mileage	Spam Scanner
Craps Game Modification	Global Warming Facts Quiz	Spelling Checker
Credit Limits	Guess the Number Game	Target-Heart-Rate Calculator
Crossword Puzzle Generator	Hangman Game	Tax Plan Alternatives; The "Fair Tax"
Cryptograms	Health Records Knight's Tour	Telephone number word generator
De Morgan's Laws	Limericks	"The Twelve Days of Christmas" Song
Dice Rolling	Maze Traversal: Generating Mazes Randomly	Tortoise and the Hare Simulation
	Morse Code	Towers of Hanoi
	Payroll System Modification	World Population Growth
	Peter Minuit Problem	
	Phishing Scanner	
	Pig Latin	
	Polymorphic Banking Program Using Account Hierarchy	

Fig. 3 | A sampling of the book's exercises.

Drawings and diagrams		
Main text drawings and diagrams		
Data hierarchy	`while` repetition statement UML activity diagram	Pass-by-value and pass-by-reference analysis of a program
Compilation and linking process for multiple source file programs	`for` repetition statement UML activity diagram	Inheritance hierarchy diagrams
Order in which a second-degree polynomial is evaluated	`do...while` repetition statement UML activity diagram	Function-call stack and activation records
`GradeBook` class diagrams	`switch` multiple-selection statement activity diagram	Recursive calls to function `fibonacci`
`if` single-selection statement activity diagram	C++'s single-entry/single-exit sequence, selection and repetition statements	Pointer arithmetic diagrams
`if...else` double-selection statement activity diagram	Graphical representation of a list	`CommunityMember` Inheritance hierarchy
`public`, `protected` and `private` inheritance	Operation `insertAtFront` represented graphically	`Shape` inheritance hierarchy
`Employee` hierarchy UML class diagram	Operation `insertAtBack` represented graphically	Operation `removeFromBack` represented graphically
How `virtual` function calls work	Operation `removeFromFront` represented graphically	Circular, singly linked list
Stream-I/O template hierarchy		Doubly linked list
Two self-referential class objects linked together		Circular, doubly linked list
		Graphical representation of a binary tree
ATM Case Study drawings and diagrams		
Use case diagram for the ATM system from the User's perspective	Classes in the ATM system with attributes and operations	Class diagram showing composition relationships of a class `Car`
Class diagram showing an association among classes	Communication diagram of the ATM executing a balance inquiry	Class diagram for the ATM system model including class `Deposit`
Class diagram showing composition relationships	Communication diagram for executing a balance inquiry	Activity diagram for a `Deposit`

Drawings and diagrams		
Class diagram for the ATM system model	Sequence diagram that models a `Withdrawal` executing	transaction Sequence diagram that models a `Deposit` executing
Classes with attributes	Use case diagram for a modified version of our ATM system that also allows users to transfer money between accounts	
State diagram for the ATM		
Activity diagram for a `Balance-Inquiry` transaction		
Activity diagram for a `Withdrawal` transaction		

Fig. 4 | A sampling of the book's drawings and diagrams.

Other Features

- ***Pointers.*** We provide thorough coverage of the built-in pointer capabilities and the intimate relationship among built-in pointers, C strings and built-in arrays.
- ***Visual presentation of searching and sorting, with a simple explanation of Big O.***
- ***Printed book contains core content; additional content is online.*** A few online chapters and appendices are included.
- ***Debugger appendices.*** We provide three debugger appendices on the book's Companion Website—Appendix H, Using the Visual Studio Debugger, Appendix I, Using the GNU C++ Debugger and Appendix J, Using the Xcode Debugger.

Secure C++ Programming

It's difficult to build industrial-strength systems that stand up to attacks from viruses, worms, and other forms of "malware." Today, via the Internet, such attacks can be instantaneous and global in scope. Building security into software from the beginning of the development cycle can greatly reduce vulnerabilities.

The CERT® Coordination Center(`www.cert.org`)was created to analyze and respond promptly to attacks. CERT—the Computer Emergency Response Team—is a government-funded organization within the Carnegie Mellon University Software Engineering Institute. CERT publishes and promotes secure coding standards for various popular programming languages to help software developers implement industrial strength systems that avoid the programming practices that leave systems open to attacks.

We'd like to thank Robert C. Seacord, Secure Coding Manager at CERT and an adjunct professor in the Carnegie Mellon University School of Computer Science. Mr. Seacord was a technical reviewer for our book, *C How to Program, 7/e*, where he scrutinized our C programs from a security standpoint, recommending that we adhere to the *CERT C Secure Coding Standard.*

We've done the same for *C++ How to Program, 9/e*, adhering to the *CERT C++ Secure Coding Standard*, which you can find at:

```
www.securecoding.cert.org
```

We were pleased to discover that we've already been recommending many of these coding practices in our books. We upgraded our code and discussions to conform to these prac tices, as appropriate for an introductory/intermediate-level textbook. If you'll be building industrial-strength C++ systems, consider reading *Secure Coding in C and C++, Second Edition* (Robert Seacord, Addison-Wesley Professional).

Online Content[①]

The book's Companion Website, which is accessible at

```
www.pearsoninternationaleditions.com/deitel
```

contains the following chapters and appendices in searchable PDF format:

- Chapter 24, C++11: Additional Features.
- Chapter 25, ATM Case Study, Part 1: Object-Oriented Design with the UML.
- Chapter 26, ATM Case Study, Part 2: Implementing an Object-Oriented Design.
- Appendix F, C Legacy Code Topics.
- Appendix G, UML 2: Additional Diagram Types.
- Appendix H, Using the Visual Studio Debugger.
- Appendix I, Using the GNU C++ Debugger.
- Appendix J, Using the Xcode Debugger.
- Appendix K, Test Driving a C++ Program on Mac OS X. (The test drives for Windows and Linux are in Chapter 1.)

The Companion Website also includes:

- Building Your Own Compiler exercise descriptions from Chapter 19 (posted at the Companion Website and at `www.deitel.com/books/cpphtp9`).
- Chapter 1 test-drive for Mac OS X.

Dependency Chart

The chart in Fig. 5 shows the dependencies among the chapters to help instructors plan their syllabi. *C++ How to Program, 9/e* is appropriate for CS1 and many CS2 courses. The chart shows the book's modular organization.

Teaching Approach

C++ How to Program, 9/e, contains a rich collection of examples. We stress program clarity and concentrate on building well-engineered software.

Live-code approach. The book is loaded with "live-code" examples—most new concepts are presented in *complete working C++ applications*, followed by one or more executions showing program inputs and outputs. In the few cases where we use a code snippet, to ensure that it's correct we tested it in a complete working program, then copied and pasted it into the book.

Syntax coloring. For readability, we syntax color all the C++ code, similar to the way most C++ integrated-development environments and code editors syntax color code. Our coloring conventions are as follows:

```
comments appear like this
keywords appear like this
constants and literal values appear like this
all other code appears in black
```

Code highlighting. We place light-blue shaded rectangles around key code segments.

Using fonts for emphasis. We emphasize on-screen components in the bold Helvetica font (e.g., the File menu) and C++ program text in the `Lucida` font (for example, `int x = 5;`).

---

① 相关资源可登录华信教育资源网(www.hxedu.com.cn)下载。

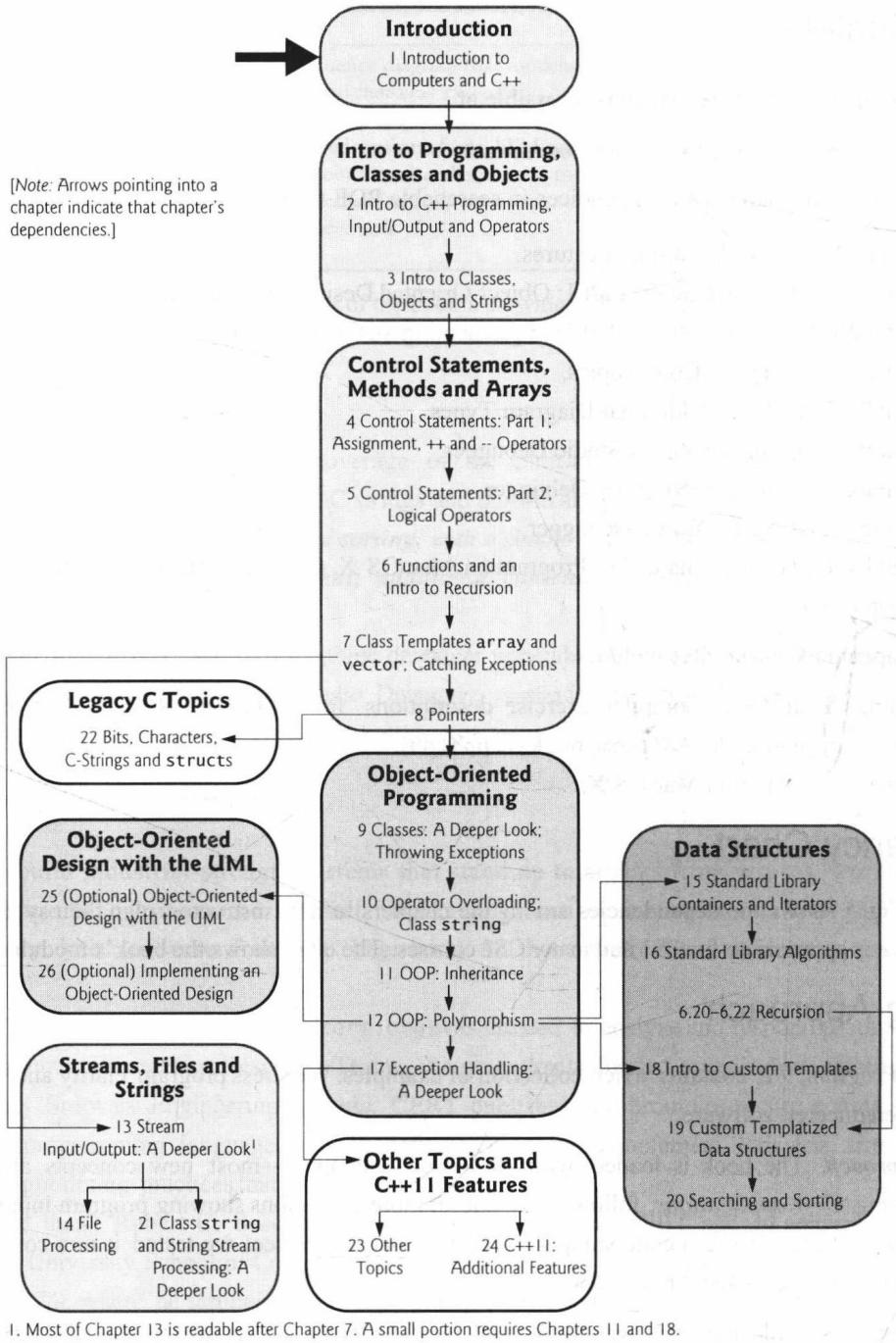


Fig. 5 | Chapter Dependency Chart.

Objectives. The opening quotes are followed by a list of chapter objectives.

Programming tips. We include programming tips to help you focus on key aspects of pro gram development. These tips and practices represent the best we've gleaned from a com bined seven decades of teaching and industry experience.



Good Programming Practices

The Good Programming Practices *call attention to techniques that will help you pro duce programs that are clearer, more understandable and more maintainable.*

需要全本请访问 www.ertongbook.com