HZ BOOKS
华章教育

PEARSON

经典原版书库

C++程序设计

（美）Y. Daniel Liang 著
阿姆斯特朗亚特兰大州立大学

（英文版·第3版）

Introduction to Programming with C++

THIRD EDITION

Y. Daniel Liang

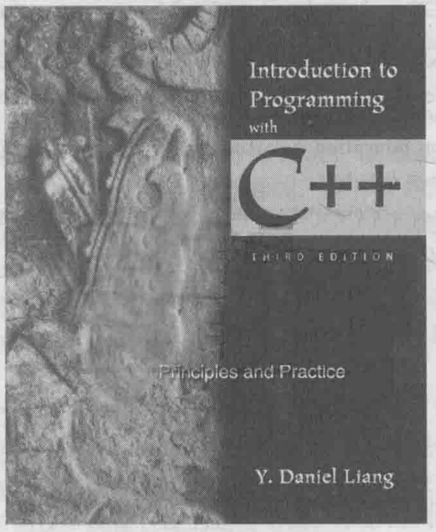
机械工业出版社
China Machine Press

经 典 原 版 书 库

C++程序设计

（英文版·第3版）

Introduction to Programming with C++ (Third Edition)



（美）Y. Daniel Liang 著
阿姆斯特朗亚特兰大州立大学

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C++程序设计（英文版·第3版）/（美）梁勇（Liang, Y. D.）著. —北京：机械工业出版社，2013.5
（经典原版书库）

书名原文：Introduction to Programming with C++，Third Edition

ISBN 978-7-111-42505-2

I. C…　II. 梁…　III. C程序－程序设计－英文　IV. TP312

中国版本图书馆CIP数据核字（2013）第100376号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问　北京市展达律师事务所

本书版权登记号：图字：01-2013-2093

Original edition, entitled: Introduction to Programming with C++, Third Edition, 9780133252811 by Y. Daniel Liang, published by Pearson Education, Inc, publishing as Pearson, Copyright © 2014, 2010, 2007.

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 reprint edition published by PEARSON EDUCATION ASIA LTD., and CHINA MACHINE PRESS, Copyright © 2013.

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).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

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

机械工业出版社（北京市西城区百万庄大街22号　　邮政编码　100037）
责任编辑：迟振春
北京瑞德印刷有限公司印刷
2013年6月第1版第1次印刷
170mm×242mm·44.25印张
标准书号：ISBN 978-7-111-42505-2
定　　价：79.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势，也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，信息学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的信息产业发展迅猛，对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，John Wiley & Sons，Elsevier，Cambridge 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出《Digital Design: Principles and Practices，4E（数字设计原理与实践，原书第 4 版）》(John F.Wakerly 著)、《Fundamentals of Digital Logic with Verilog Design（数字逻辑基础与 Verilog 设计）》(Stephen Brown 著)、《Electromagnetic Field Theory Fundamentals，2E（电磁场与电磁波，原书第 2 版）》(Bhag Singh Guru 著)、《Fundamentals of Electric Circuits，5E(电路基础，原书第 5 版、英文版第 5 版)》(Charles K. Alexander 著)、《Digital Fundamentals: A Systems Approach（数字基础：系统方法）》(Thomas L. Floyd 著)、《Introductory Circuit Analysis ,12E(电路分析导论，原书第 12 版，本科教学版)》(Robert L.Boylestad 著)、《Foundations of MEMS，2E (微机电系统基础（原书第 2 版)》(Chang Liu 著）等大师名家的经典教材，以“国外电子电气经典教材系列”为总称出版，供读者学习、研究及珍藏。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着电子电气专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外电子电气教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com

电子邮件：hzjsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街 1 号

邮政编码：100037

HZ BOOKS
华章教育

华章科技图书出版中心

This book is dedicated to my current and former C++ students. You have inspired and helped me to continue to improve this book.

To Samantha, Michael, and Michelle

此为试读,需要完整PDF请访问: www.ertongbook.com

DEAR READER,

Many of you have provided feedback on previous editions of *Introduction to Programming with C++*, and your comments and suggestions have greatly improved the book. This edition has been substantially enhanced in presentation, organization, examples, exercises, and supplements—including the following:

- Reorganized sections and chapters present subjects in a logical order
- Many new interesting examples and exercises stimulate interest
- Introduction of the `string` type in Chapter 4 enables students to write programs using strings early
- Key Points at the beginning of each section highlight important concepts and materials
- Check Points at the end of each section verify the student's understanding of the material covered

what's new?

Please visit www.cs.armstrong.edu/liang/cpp3e/correlation.html for a complete list of new features as well as correlations to the previous edition.

This book teaches programming using a problem-driven method that focuses on problem solving rather than syntax. We make introductory programming interesting by using thought provoking problems in a broad context. The central thread of early chapters is on problem solving. Appropriate syntax and libraries are introduced to enable readers to write programs to solve problems. To support the teaching of programming in a problem-driven way, the book provides a wide variety of problems at various levels of difficulty to motivate students. To appeal to students in all majors, the problems cover many application areas, including math, science, business, finance, gaming, and animation.

problem driven

The book focuses on fundamentals first by introducing basic programming concepts and techniques before designing custom classes. The fundamental concepts and techniques of loops, functions, and arrays are the basis for programming. Building this strong foundation prepares students to learn object-oriented programming and advanced C++ programming.

fundamentals first

This book teaches C++. The fundamentals of problem solving and programming are the same regardless of which programming language you use. You can learn programming using any high-level programming language such as Python, Java, C++, or C#. Once you know how to program in one language, it is easy to pick up other languages, because the basic techniques for writing programs are the same.

examples and exercises

The best way to teach programming is *by example*, and the only way to learn programming is *by doing*. Basic concepts are explained by example and many exercises with various levels of difficulty are provided for students to practice. In our programming courses, we assign programming exercises after each lecture.

Our goal is to produce a text that teaches problem solving and programming in a broad context using a wide variety of interesting examples. If you have any comments or suggestions for improving this book, please email me.

Sincerely,

Y. Daniel Liang
y.daniel.liang@gmail.com
www.cs.armstrong.edu/liang
www.pearsonhighered.com/liang

PREFACE

What's New in This Edition?

This third edition substantially improves *Introduction to Programming with C++, Second Edition*. The major improvements are as follows:

complete revision

- A complete revision to enhance clarity, presentation, content, examples, and exercises

new examples and exercises

- New examples and exercises to motivate and stimulate student interest in programming

Key Points

- Key Points that highlight the important concepts covered in each section

Check Points

- Check Points that provide review questions to help students track their learning progress and evaluate their knowledge about a major concept or example

VideoNotes

- New VideoNotes that provide short video tutorials designed to reinforce the key concepts

string objects early

- Introduction of `string` objects in Chapter 4 to enable strings to be used in the early part of the book

simple IO early

- Introduction of simple input and output in Chapter 4 to enable students to write programs using files early

functions in one chapter

- Inclusion of functions in Chapter 6, which now covers all issues related to functions

common error sections

- Chapter sections on common errors and pitfalls to steer students away from common programming errors

simplified examples

- Replacement of complex examples with simpler ones (e.g., Solving the Sudoku problem in Chapter 8 is replaced by a problem of checking whether a solution is correct. The complete solution to the Sudoku problem is moved to the Companion Website.)

algorithm efficiency and techniques

- Expanded bonus Chapter 18 introduces algorithmic techniques: dynamic programming, divide-and-conquer, backtracking, and greedy algorithm with new examples to design efficient algorithms

C++11

- Introduction of new C++11 features of foreach loops and auto type inference in the bonus chapters and of lambda functions in the supplements on the Companion Website

Pedagogical Features

The book uses the following elements to help students gain the most from the material:

- The chapter **Objectives** list what students should learn so that they can determine whether they have met these objectives after completing the chapter.
- The chapter **Introduction** opens the discussion with representative problems to give the reader an overview of what to expect.
- **Key Points** highlight the important concepts covered in each section.
- **Check Points** provide review questions to help students track their progress as they read the chapter and evaluate their learning.
- **Problems and Case Studies**, carefully chosen and presented in an easy-to-follow style, teach problem solving and programming concepts. The book uses many small, simple, and stimulating examples to present important ideas.
- The **Chapter Summary** reviews the important subjects that students should understand and remember. It helps them reinforce the key concepts of the chapter.

- Self-test quizzes are available online through MyProgrammingLab (www.myprogramminglab .com) for students to self-test on programming concepts and techniques.
- **Programming Exercises**, grouped by sections, provide students with opportunities to apply their newly acquired skills. The level of difficulty is rated as easy (no asterisk), moderate (*), hard (**), or challenging (***). The trick of learning programming is practice, practice, and practice. To that end, the book provides numerous exercises.
- **Notes**, **Tips**, **Cautions**, and **Pedagogical Notes** are inserted throughout the text to offer valuable advice and insight on important aspects of program development.

Note
Provides additional information on the subject and reinforces important concepts.

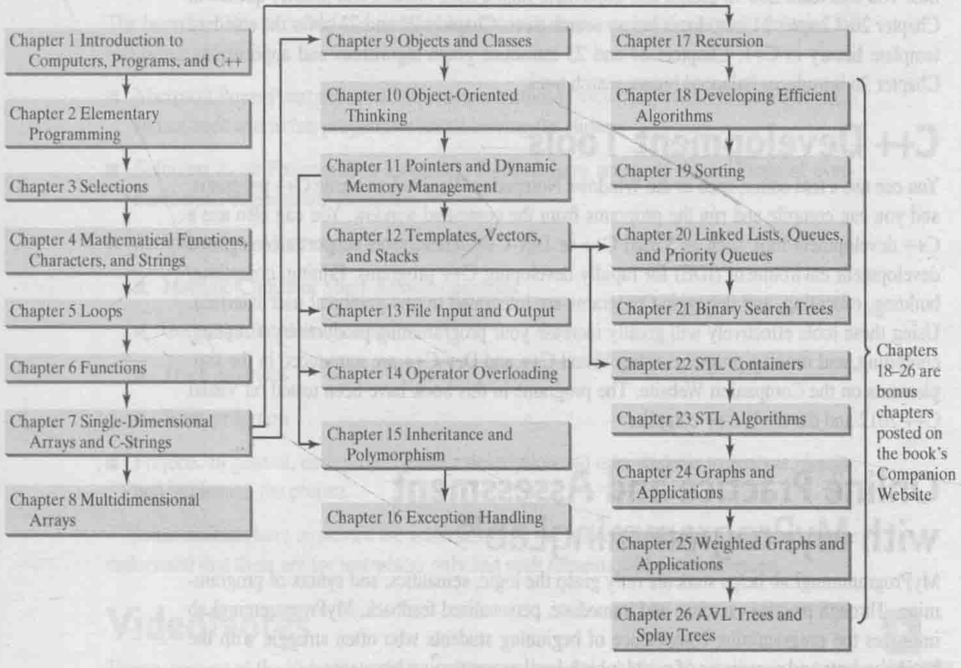
Tip
Teaches good programming style and practice.

Caution
Helps students avoid the pitfalls of programming errors.

Pedagogical Note
Gives advice on how to use the materials in the book effectively.

Flexible Chapter Orderings

The book provides flexible chapter orderings, as shown in the following diagram:



Chapter 1 Introduction to Computers, Programs, and C++
Chapter 2 Elementary Programming
Chapter 3 Selections
Chapter 4 Mathematical Functions, Characters, and Strings
Chapter 5 Loops
Chapter 6 Functions
Chapter 7 Single-Dimensional Arrays and C-Strings
Chapter 8 Multidimensional Arrays
Chapter 9 Objects and Classes
Chapter 10 Object-Oriented Thinking
Chapter 11 Pointers and Dynamic Memory Management
Chapter 12 Templates, Vectors, and Stacks
Chapter 13 File Input and Output
Chapter 14 Operator Overloading
Chapter 15 Inheritance and Polymorphism
Chapter 16 Exception Handling
Chapter 17 Recursion
Chapter 18 Developing Efficient Algorithms
Chapter 19 Sorting
Chapter 20 Linked Lists, Queues, and Priority Queues
Chapter 21 Binary Search Trees
Chapter 22 STL Containers
Chapter 23 STL Algorithms
Chapter 24 Graphs and Applications
Chapter 25 Weighted Graphs and Applications
Chapter 26 AVL Trees and Splay Trees
Chapters 18–26 are bonus chapters posted on the book's Companion Website

Organization of the Book

The chapters can be grouped into three parts, which together form a solid introduction to problem solving and programming using C++.

Part I: Fundamentals of Programming (Chapters 1–8)

This part is a stepping-stone, which prepares you to embark on the journey of learning programming with C++. You will begin to know C++ (Chapter 1) and will learn elementary programming techniques with primitive data types, expressions, and operators (Chapter 2), selection statements (Chapter 3), mathematical functions, characters, and strings (Chapter 4), loops (Chapter 5), functions (Chapter 6), and arrays (Chapters 7–8).

Part II: Object-Oriented Programming (Chapters 9–16)

This part introduces object-oriented programming. C++ is an object-oriented programming language that uses abstraction, encapsulation, inheritance, and polymorphism to provide great flexibility, modularity, and reusability in developing software. You will learn programming with objects and classes (Chapter 9); design classes (Chapter 10); explore pointers and dynamic memory management (Chapter 11); develop generic classes using templates (Chapter 12); use IO classes for file input and output (Chapter 13); use operators to simplify functions (Chapter 14); define classes from base classes (Chapter 15); and create robust programs using exception handling (Chapter 16).

Part III: Algorithms and Data Structures (Chapter 17 and Bonus Chapters 18–26)

This part introduces the main subjects in a typical data structures course. Chapter 17 introduces recursion to write functions for solving inherently recursive problems. Chapter 18 introduces how to measure algorithm efficiency in order to choose an appropriate algorithm for applications. Chapter 19 presents various sorting algorithms and analyzes their complexities. You will learn how to design and implement linked lists, queues, and priority queues in Chapter 20. Chapter 21 introduces binary search trees. Chapters 22 and 23 cover the standard template library in C++. Chapters 24 and 25 introduce graph algorithms and applications. Chapter 26 introduces balanced binary search trees.

C++ Development Tools

You can use a text editor, such as the Windows Notepad or WordPad, to create C++ programs, and you can compile and run the programs from the command window. You can also use a C++ development tool, such as Visual C++ or Dev-C++. These tools support an integrated development environment (IDE) for rapidly developing C++ programs. Editing, compiling, building, executing, and debugging programs are integrated in one graphical user interface. Using these tools effectively will greatly increase your programming productivity. Creating, compiling, and running programs using Visual C++ and Dev-C++ are introduced in the supplements on the Companion Website. The programs in this book have been tested on Visual C++ 2012 and the GNU C++ compiler.

Online Practice and Assessment with MyProgrammingLab™

MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students who often struggle with the basic concepts and paradigms of popular high-level programming languages.

A self-study and homework tool, a MyProgrammingLab course consists of hundreds of small practice problems organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong—and why. For instructors, a comprehensive gradebook tracks correct and incorrect answers and stores the code inputted by students for review.

MyProgrammingLab is offered to users of this book. For a full demonstration, to see feedback from instructors and students, or to get started using MyProgrammingLab in your course, visit www.myprogramminglab.com.

Student Resource Website

The Student Resource Website, accessible from www.pearsonhighered.com/liang, contains the following:

- Answers to Check Points
- Solutions to even-numbered Programming Exercises
- Source code for the examples
- Algorithm animations
- Errata

Supplements

The text covers the essential subjects. The supplements extend the text to introduce additional topics that might be of interest to readers. The supplements are available on the Companion Website (www.pearsonhighered.com/liang).

Instructor Resource Website

The Instructor Resource Website, accessible from www.pearsonhighered.com/liang, contains the following:

- Microsoft PowerPoint slides with interactive buttons to view full-color, syntax-highlighted source code and to run programs without leaving the slides.
- Solutions to all Programming Exercises. Students have access to the solutions of even-numbered Programming Exercises.
- Sample exams. Most exams have four parts:
  - Multiple-Choice or Short-Answer questions
  - Correct programming errors
  - Trace programs
  - Write programs
- Projects. In general, each project gives a description and asks students to analyze, design, and implement the project.

Some students have requested the materials from the Instructor Resource Website. Please understand that these are for instructors only and such requests will not be honored.

VideoNotes

VideoNote

Twenty percent of the VideoNotes in this edition are brand new! VideoNotes were introduced in the previous edition to provide additional help by presenting examples of key topics and to show how to solve problems completely, from design through coding. VideoNotes can be accessed on the book's Companion Website using the student access code printed on the inside front cover of this book. If you have a used book, you can purchase access to the VideoNotes and other premium content through the Purchase link on the Companion Website (www.pearsonhighered.com/liang).

Acknowledgments

I would like to thank Armstrong Atlantic State University for enabling me to teach what I write and for supporting me to write what I teach. Teaching is the source of inspiration for continuing to improve the book. I am grateful to the instructors and students who have offered comments, suggestions, bug reports, and praise.

This book was greatly enhanced thanks to outstanding reviews for this and previous editions. The following reviewers contributed: Anthony James Allevato (Virginia Tech); Alton B. Coalter (University of Tennessee, Martin); Linda Cohen (Forsyth Tech); Frank David Ducrest (University of Louisiana, Lafayette); Waleed Farag (Indiana University of Pennsylvania); Max I. Fomitchev (Penn State University); Jon Hanrath (Illinois Institute of Technology); Michael Hennessy (University of Oregon); Debbie Kaneko (Old Dominion University); Henry Ledgard (University of Toledo); Brian Linard (University of California, Riverside); Dan Lipsa (Armstrong Atlantic State University); Jayantha Herath (St. Cloud State University); Daqing Hou (Clarkson University); Hui Liu (Missouri State University); Ronald Marsh (University of North Dakota); Peter Maurer (Baylor University); Jay McCarthy (Brigham Young University); Jay D. Morris (Old Dominion University); Charles Nelson (Rock Valley College); Ronald Del Porto (Pennsylvania State University); Mitch Pryor (University of Texas); Martha Sanchez (University of Texas at Dallas); William B. Seales (University of Kentucky); Kate Stewart (Tallahassee Community College); Ronald Taylor (Wright State University); Matthew Tennyson (Bradley University); David Topham (Ohlone College); Margaret Tseng (Montgomery College); and Barbara Tulley (Elizabethtown College).

It is a great pleasure, honor, and privilege to work with Pearson. I would like to thank Tracy Johnson and her colleagues Marcia Horton, Carole Snyder, Yez Alayan, Scott Disanno, Kayla Smith-Tarbox, Gillian Hall, and their colleagues for organizing, producing, and promoting this project.

As always, I am indebted to my wife, Samantha, for her love, support, and encouragement.

此为试读，需要完整PDF请访问： www.ertongbook.com

C++ Quick Reference

if Statements

```
if (condition)
{
  statements;
}

if (condition)
{
  statements;
}
else
{
  statements;
}

if (condition1)
{
  statements;
}
else if (condition2)
{
  statements;
}
else
{
  statements;
}
```

switch Statements

```
switch (intExpression)
{
  case value1:
    statements;
    break;
  ...
  case valuen:
    statements;
    break;
  default:
    statements;
}
```

Array/Initializer

```
int list[10];
int list[] = {1, 2, 3, 4};
```

Multidimensional Array/Initializer

```
int list[10][15];
int list[2][2] = {{1, 2}, {3, 4}};
```

Dynamic Memory Creation/Deletion

```
int* p1 = new int;
int* p2 = new int[10];
delete p1;
delete [] p2;
```

Loop Statements

```
while (condition)
{
  statements;
}

do
{
  statements;
} while (condition);

for (init; condition;
  adjustment)
{
  statements;
}
```

Frequently Used functions

time(0)	returns current time
srand(seed)	sets a new seed for generating random numbers
rand()	returns a random integer
pow(a, b)	returns a^b

Character Functions

isdigit(c)	returns true if c is a digit.
isalpha(c)	returns true if c is a letter.
isalnum(c)	returns true if c is a letter or a digit.
islower(c)	returns true if c is a lowercase letter.
isupper(c)	returns true if c is an uppercase letter.
tolower(c)	returns a lowercase for c.
toupper(c)	returns an uppercase for c.

C-String Functions

strlen	returns string length
strcpy	copies a string
strcat	concatenate two strings
strcmp	compares two strings
atol	converts a string to a long value
itoa	converts a an integer a string

The string Class Member Functions

append	appends new contents to the string
insert	inserts new contents to the string
at	retrieves character from the string
[]	string subscript operator
length	returns the length of the string.
substr	returns a substring from the string

Companion Website: www.cs.armstrong.edu/liang/cpp3e

CONTENTS

此为试读，需要完整PDF请访问：www.ertongbook.com