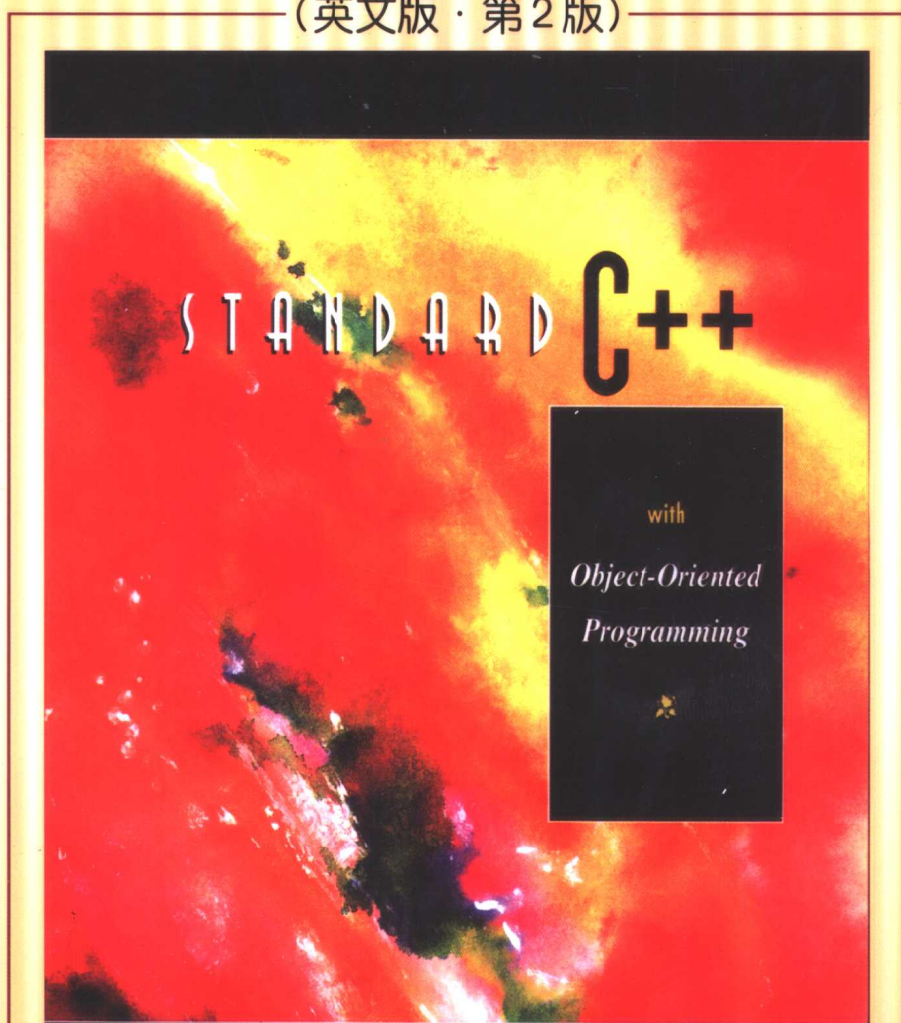


经 典 原 版 书 库

# 标准 C++ 与面向对象 程序设计

—(英文版·第2版)—



PAUL

(美) Paul S. Wang 著  
肯特州立大学



机械工业出版社  
China Machine Press

THOMSON

经典原版书库

TP312  
Y184

# 标准 C++ 与面向对象 程序设计

(英文版·第2版)

Standard C++ with Object-Oriented Programming  
(Second Edition)

(美) Paul S. Wang 著  
肯特州立大学

北方工业大学图书馆



00730490



机械工业出版社  
China Machine Press

Paul S. Wang: Standard C++ with Object-Oriented Programming, Second Edition (ISBN: 0-534-37131-0).

Original edition copyright © 2001 by Brooks/Cole.

First published by Brooks/Cole, an imprint of Thomson Learning, United States of America.

All rights reserved.

Reprinted for the People's Republic of China by Thomson Asia Pte Ltd and China Machine Press under the authorization of Thomson Learning. No part of this book may be reproduced in any form without the express written permission of Thomson Learning Asia and China Machine Press.

本书英文影印版由汤姆森学习出版社与机械工业出版社合作出版。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。

版权所有,侵权必究。

**本书版权登记号: 图字: 01-2003-4665**

### **图书在版编目 ( CIP ) 数据**

标准C++与面向对象程序设计 ( 英文版·第2版) / ( 美 ) 王士弘 ( Wang, P. S. ) 著. -北京: 机械工业出版社, 2003. 7

( 经典原版书库 )

书名原文: Standard C++ with Object-Oriented Programming, Second Edition

ISBN: 7-111-12542-8

I. 标… II. 王… III. C++语言-程序设计-英文 IV. TP312

中国版本图书馆CIP数据核字 ( 2003 ) 第055262号

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: 杨海玲

北京瑞德印刷有限公司印刷·新华书店北京发行所发行

2003年7月第1版第1次印刷

787mm×1092mm 1/16·37 印张

印数: 0 001 - 3 000册

定价: 55.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

# 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中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，计算机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，还揭橥了学术的源变，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，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机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、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及收藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方式如下：

电子邮件：[hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元  
石教英  
张立昂  
邵维忠  
周克定  
郑国梁  
高传善  
裘宗燕

王 珊  
吕 建  
李伟琴  
陆丽娜  
周傲英  
施伯乐  
梅 宏  
戴 葵

冯博琴  
孙玉芳  
李师贤  
陆鑫达  
孟小峰  
钟玉琢  
程 旭

史忠植  
吴世忠  
李建中  
陈向群  
岳丽华  
唐世渭  
程时端

史美林  
吴时霖  
杨冬青  
周伯生  
范 明  
袁崇义  
谢希仁

To my mother, Chang Pei Wang,  
on her 80th birthday:

**Happy Birthday, Mom. . .**

# Preface



**C**++ is one of the most popular languages for object-oriented programming (OOP), the centerpiece of object technology (OT) that is revolutionizing the software industry. OOP creates programs that are well organized, easy to understand and modify, flexible, and reusable in many different situations. It reduces complexity and makes software production and maintenance more economical. C++ has been standardized jointly by the International Standards Organization (ISO) and the American National Standards Institute (ANSI). The ISO/ANSI C++ standard (ISO/IEC FDIS 14882) was approved in November 1997.

New features introduced in Standard C++ include the Boolean type `bool`, exceptions, namespaces, run-time type identification, type cast notations, and template libraries with generic algorithms. Standard C++ also revised and extended a number of existing features including the wide character type, templates, and function call resolution. A new `string` class makes character strings easier to use.

This book is a revision of *C++ with Object-Oriented Programming*, covering Standard C++ comprehensively while preserving the proven approach. The C++ constructs are matched with clear and precise OOP concepts. Language mechanisms are explained individually and in combination to achieve OOP objectives. Object-based, object-oriented, and generic programming techniques are demonstrated in realistic programs to show their applications in practice. Chapter 12 shows how C++ can be applied for Web CGI programming.

## OBJECT ORIENTATION

C++ is important because it is efficient and supports OOP well. Thus, OOP concepts and techniques are taught as an integral part of C++ programming. The approach brings OOP concepts down to earth so that they are easily grasped. Object orientation is introduced early and demonstrated with many complete examples. How these techniques are applied to solve problems and how they make programs more flexible and reusable are clearly shown.



Key OOP concepts such as data abstraction, encapsulation, information hiding, problem solving with objects, genericness, inheritance, and polymorphism are presented clearly and comprehensively. Basics on object-oriented design are also included. Again, these are illustrated by many C++ examples, including a bank account example that is carried through many chapters.

## HANDS-ON APPROACH

The best way to learn programming is to write programs. With clear concepts and good examples, the text encourages the writing of interesting programs early. Chapters 1 and 2 form a primer introducing essential components of C++ and object-based programming to get started quickly. Materials on thinking with objects, C++ programming tips, and style guides help beginning programmers.

A pocket calculator simulation program is introduced in Chapter 5. This program evolves with each new chapter through programming exercises. Finally, in Chapter 13, a substantial pocket calculator program emerges that connects many key OOP concepts and C++ constructs.

## COMPREHENSIVE COVERAGE

Standard C++ is presented comprehensively and in depth. The coverage anticipates the needs of students and describes Standard C++ as an integral, self-contained language. The examples evolve as new material is covered, providing many chances to revisit familiar code and to focus on the new concepts and features being introduced.

It is possible to use this book as the sole text for a course. Besides the basic topics, it covers the Standard Library, the I/O stream classes, templates, the Standard Template Library (STL), generic algorithms, program organization, use of header files, error and exception handling, preprocessing, and compilation (on UNIX<sup>®</sup> and on PC).

Object orientation is emphasized with topics on building software objects, comparing external behavior versus internal workings, reducing complexity with encapsulation, deriving new classes based on existing ones, writing generic codes that are reusable in many situations, creating software black boxes that are plug-compatible in usage, and establishing polymorphic procedures and objects that work with multiple types of objects. Also included are object-oriented design techniques, methods, and applications.

## OOP MADE EASY

Standard C++ is a large and complex language. It is easy to get lost in the maze of new OOP concepts and the supporting C++ constructs. No effort is spared to make this complicated subject easy to grasp and understand. The approach begins with simple topics and key concepts as a foundation. Then, advanced topics are added in a logical sequence that is easy to follow. The materials are organized to anticipate questions and provide answers. Clear, interesting, and realistic examples show how to write object-oriented programs and how to apply the concepts and techniques introduced.

Inheritance, a key OOP feature, tends to be difficult for beginning programmers. A clear mental picture is painted of a derived class and its relation with a base class upon which it is built. Furthermore, specific principles on class derivation and its proper usage are supplied. Multiple inheritance is also covered with clarity and good examples.

Polymorphism and plug compatibility are central OOP techniques that require sophistication to use well. An entire chapter is devoted to this subject that provides enough material to challenge even the most advanced students.

## WEB APPLICATION

CGI programs written in C++ can be fast and efficient. HTML forms and CGI programs for form processing are presented in a concise chapter that puts the OOP concepts and C++ techniques to good use. A C++ class library for CGI programming is given and explained. CGI programs using this library for form processing are presented.

## FLEXIBLE USAGE

The book contains more than enough material for a three-credit programming course at the junior, senior, or beginning graduate level. No C background is assumed, but appropriate programming experience, to appreciate software complexity, is highly recommended. Knowledge of C or ANSI C will reduce the amount of work involved. At a lower level, the material in the beginning chapters should be covered carefully. Advanced topics such as user-defined free storage management (Section 8.12), CGI programming (Chapter 12), and object-oriented design (Chapter 13) can be omitted. At a higher level, Chapter 1 and Chapter 14 can be assigned to students as background reading to allow more class time for other topics.

For a class with substantial programming experience, generic programming, template writing, and object-oriented programming projects, as suggested by the end-of-chapter exercises, can be emphasized. In this case, the design considerations covered in Chapter 13 may be discussed much earlier.

The preprocessing and compilation material in Chapter 14 can be introduced (or assigned for reading by students) whenever the instructor feels it is appropriate in a course. Chapters 2 through 11 are central to this text and should prove challenging and rewarding to any reader.

The book can also be used as a valuable supplement to a general course on OOP, data structures, or object-oriented design with C++ implementation.

## RIGHT FOR YOUR SYSTEM

C++ is presented in a system-independent manner. The examples will run on any reasonable C++ implementation including workstations, PCs, and multi-user servers. Chapter 14 gives general information on preprocessing, compiling, and executing programs. The Free Software Foundation offers g++, a good implementation of C++ for free downloading:

<http://www.gnu.ai.mit.edu/software/gcc> (for UNIX)

<http://www.delorie.com/djgpp/> (for PC)

## EASY REFERENCE

As an instructional guide, this text follows an incremental approach, whereby new concepts are built on old ones to make understanding easy. However, the book is also a valuable reference tool. Information has been organized for easy reference with tables, figures, displayed syntax explanations, examples, and summaries. All key C++ constructs are collected in Appendix A for quick review. Appendix B summarizes usage of special member functions. Other appendices cover debugging, library functions, and the mixed use of C++ and C. Accurate cross-referencing and a comprehensive index help locate information contained in the book.

## EXAMPLE PACKAGE

Throughout the text, concepts and programming constructs are amply illustrated with examples of practical importance. The example package is organized by chapter. There are about 400 files containing complete source codes

that are ready to compile. The complete example package for UNIX and for PC is available from the Brooks/Cole Web site

[www.brookscole.com](http://www.brookscole.com)

## ACKNOWLEDGMENTS

The book has benefited from the previous edition, extensive classroom use, and feedback from industry. In particular, I wish to thank W. B. Adams of Goodyear Tire and Rubber Company, Technical Computer Operations (Akron, Ohio). The book has gone through several rounds of reviews, and I would like to thank the following reviewers for their suggestions and corrections:

William G. Albrecht	Florida Southern College
Richard Chang	University of Maryland
Julius Dichter	University of Bridgeport
Craig A. Adams	Houston Community College System
Hristina Galabova	Home Box Office, Inc.

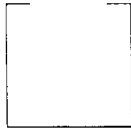
Especially, I'd like to thank Hristina Galabova, a Kent State alumna, for her thorough review and thoughtful suggestions, resulting in many improvements.

The final draft of the book went through classroom trial at Kent State University in the spring of 2000. This resulted in many improvements. My appreciation goes to all the students who took the Object-Oriented Programming class. Thanks also go to Joyce Fuell, my assistant at the Institute for Computational Mathematics of the Department of Mathematics and Computer Science at Kent State University, for helping with the production of the manuscript. I would like to thank Kallie Swanson, the editor at Brooks/Cole for her expert handling of this publication, to Merrill Peterson at Matrix Productions for managing the production process, and to Arthur Ogawa at T<sub>E</sub>X Consultants for the L<sup>A</sup>T<sub>E</sub>X style file, graphical art, index generation, and great composition.

My daughter Laura suggested the term *appendant*, and it proved to be very helpful in describing class derivation. She deserves all the credit. Finally, I'd like to thank my wife, Jennifer, and younger children, Deborah and David, for their understanding, support, and encouragement.

Paul S. Wang  
Kent State University

# Contents



## Introduction 1

## ONE C++ Primer Part I: Basics 8

- 1.1 Object-Oriented Program Structure 8
- 1.2 Functions 9
- 1.3 Running the First Example 12
- 1.4 Statements 12
- 1.5 The `while` Statement 13
- 1.6 Simple Conditional Statements 16
- 1.7 Characters and Character Input and Output 17
- 1.8 More Basic Constructs 20
- 1.9 Data Types and Declarations 26
- 1.10 Operators and Expressions 29
- 1.11 Enumerations 35
- 1.12 Arrays 37
- 1.13 Pointers 40
- 1.14 An Example 41
- 1.15 Iteration Control 42
- 1.16 Choices by `switch` 46
- 1.17 Summary 48
- Exercises 48

## TWO C++ Primer Part II: Object-Based Programming 51

- 2.1 Data Abstraction and Encapsulation 51
- 2.2 Function Calls and Argument Passing 58
- 2.3 Command-Line Arguments 61
- 2.4 Environment Variables 62

2.5	The <code>string</code> Class	63
2.6	Problem Solving with Objects	65
2.7	C++ I/O Streams	70
2.8	Error Handling	70
2.9	Object-Based Thinking	73
2.10	C++ Programming Tips	74
2.11	Summary	77
	Exercises	78

### **THREE Key Constructs 80**

3.1	Identifier Scoping	81
3.2	Namespaces	84
3.3	More on Declarations	86
3.4	Recursion	89
3.5	A Class of Fractions	93
3.6	Optional and Variable-Length Arguments	97
3.7	Overloading Functions	99
3.8	References	102
3.9	Read-Only Variables and Parameters	103
3.10	The <code>typedef</code> Declaration	105
3.11	Storage Allocation and Management	106
3.12	A Circular Buffer	109
3.13	Implicit Type Conversions	114
3.14	Explicit Type Cast	116
3.15	How to Use Header Files	117
3.16	Summary	119
	Exercises	120

### **FOUR Arrays, Pointers, and Generic Code 123**

4.1	Array Concepts and Declarations	123
4.2	Pointers and Address Arithmetic	125
4.3	Two-Dimensional Arrays	133
4.4	A Matrix Class	135
4.5	A Class of Polynomials	138

4.6	Array Objects: vector	141
4.7	Sorting Text Lines with Objects	143
4.8	Pointers and Function Calls	151
4.9	Arrays, Pointers, and References	154
4.10	Multiple Indirection	154
4.11	Generic Programs	156
4.12	A Generic Sorting Program	162
4.13	Pointers and Dynamically Allocated Storage	165
4.14	Summary	168
	Exercises	169
<b>FIVE</b>	<b>Classes and Objects</b>	<b>172</b>
5.1	Defining Classes	172
5.2	Creating and Initializing Objects	175
5.3	The Host Object	181
5.4	Internal-External Decoupling	182
5.5	Pocket Calculator Simulation	183
5.6	Built-in Operations for Objects	192
5.7	Object Destruction	193
5.8	Friends of a Class	195
5.9	Recursive Structures	197
5.10	Instance and Static Members	204
5.11	The Size of Objects	210
5.12	Generic Lists	211
5.13	Unions and Bit Fields	216
5.14	Summary	216
	Exercises	217
<b>SIX</b>	<b>I/O Streams and the Standard Library</b>	<b>219</b>
6.1	Different Header Files	219
6.2	C-Style String Input and Output	220
6.3	String Objects	222
6.4	A String Tokenizer	225
6.5	Operations on Characters	228

6.6	Numeric Computations	229
6.7	The I/O Stream Library	230
6.8	Stream I/O for Objects	240
6.9	I/O Manipulators	242
6.10	Summary	244
	Exercises	245

## **SEVEN Inheritance and Class Derivation 248**

7.1	Class Derivation	249
7.2	Derived Classes and Objects	252
7.3	Public Derivation	255
7.4	Derivation Principles	258
7.5	Access Control Under Class Derivation	259
7.6	Private Derivation	261
7.7	Specialization of Generic Classes	263
7.8	Type Relations Under Inheritance	265
7.9	Assignment of Objects	267
7.10	Copying of Objects	271
7.11	Derived-Object Assignment and Copying	274
7.12	Object Assignment and Copying via Derivation	276
7.13	Multiple Inheritance	278
7.14	The I/O Stream Class Hierarchy	282
7.15	Summary	283
	Exercises	284

## **EIGHT Operator Overloading, Iterators, and Exceptions 287**

8.1	Operator Overloading	287
8.2	A Reference-Count String Class	290
8.3	Overloading []	293
8.4	Overloading ++ and --	294
8.5	Smart Pointers	295
8.6	Iterators	296
8.7	Nested and Local Classes	299



8.8	Robust Classes	302
8.9	User-Defined Type Conversions	303
8.10	Error and Exception Handling	306
8.11	Matrix with Exceptions	314
8.12	Managing Free Storage	316
8.13	Summary	319
	Exercises	320
<b>NINE</b>	<b>OOP Techniques</b>	<b>322</b>
9.1	Programming with Plug-Compatible Objects	322
9.2	Run-Time Type Identification	330
9.3	A Generic Tree	335
9.4	Planning Uniform Public Interfaces	343
9.5	Destruction of Plug-Compatible Objects	352
9.6	Ordering Text Lines	354
9.7	Interfaces	357
9.8	Understanding Virtual Functions	359
9.9	Inheritance Planning	361
9.10	Object-Family Classes	363
9.11	Summary	371
	Exercises	372
<b>TEN</b>	<b>Templates</b>	<b>375</b>
10.1	Basic Concepts	375
10.2	Function Templates	377
10.3	Class Templates	383
10.4	Template Compilation	388
10.5	A List Template	390
10.6	Class Template Specialization	397
10.7	Derived-Class Template	397
10.8	A Generic Hash Table	400
10.9	Generic Programming Approaches	407
10.10	Summary	409
	Exercises	409