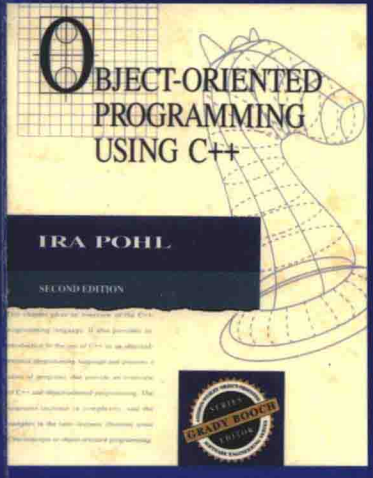
国外计算机科学教材系列

C++面向对象编程

（第二版）

Object-Oriented Programming Using C++

Second Edition



英文版

[美] Ira Pohl 著

PEARSON
Addison
Wesley

PHEI

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

经典教材

C++面向对象编程（第二版）

英文版

Object-Oriented Programming Using C++, Second Edition

本书面向有经验的编程人员，清晰、透彻地介绍了ANSI C++面向对象编程。书中讲述了支持面向对象编程概念的C++语言特性，包括STL、名称空间、RTTI以及布尔类型等新特性。Ira Pohl是C++方面的权威作家，在本书中以其著名的“剖析”方法展示了关键编程要素及惯用语言，教你如何权衡以及做出最恰当的选择。

本书主要特点

- 反映了C++ ANSI标准的最新进展，包括对新增STL库的详细论述
- 解释语言特性如何支持面向对象编程概念
- 借助示例，利用交互式的练习帮助你理解面向对象的关键概念并加以实践
- 可从网上（http://www.awl.com/cseng/titles/0-201-89550-1）下载书中所有示例程序代码，以及用于展示要点的附加程序代码

Ira Pohl

美国加州大学Santa Cruz校区计算机科学系的教授。他在软件方法学方面有二十多年的经验，是C++和C语言编程的国际权威。Ira Pohl曾兼任数字设备公司、苹果、斯坦福线性加速器中心、Xylinx、National Technological大学和Gupta的顾问。

ISBN 7-5053-9713-3

9 787505 397132 >

限中国大陆地区销售

PEARSON Addison Wesley

PHEI

责任编辑：史　平
封面设计：毛惠庚

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书

ISBN 7-5053-9713-3　定价：49.00元

PEARSON Addison Wesley

国外计算机科学教材系列

C++面向对象编程（第二版）

Object-Oriented Programming Using C++, Second Edition

英文版

電子

国外计算机科学教材系列

C++ 面向对象编程

（第二版）

（英文版）

Object-Oriented Programming Using C++
Second Edition

［美］ Ira Pohl 著

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

内容简介

本书旨在介绍使用ANSI C++进行面向对象的编程，解释在此环境中的C++特性。书中提供STL、名称空间、RTTI以及布尔类型等C++最新特性的快速指南，借助大量示例展示优秀的编程风格。全书重点介绍了C++的数据结构、标准模板库以及C++语言的主流方向和习惯用法。具体包括基本类型和语句、功能和指示器、数据隐藏、多态性、迭代器和容器、继承等多项内容。

本书面向有编程经验的学生和其他读者，可作为应用C++语言讲授的高级编程、数据结构、软件设计方法学等课程的教材。

版权贸易合同登记号　图字：01-2003-6237

图书在版编目（CIP）数据

C++面向对象编程 = Object-Oriented Programming Using C++：第二版 /（美）波尔（Pohl, I.）著.
-北京：电子工业出版社，2004.4
（国外计算机科学教材系列）
ISBN 7-5053-9713-3

Ⅰ.C… Ⅱ.波… Ⅲ.C语言-程序设计-英文 Ⅳ.TP312

中国版本图书馆CIP数据核字（2004）第015386号

责任编辑：史　平
印　　刷：北京东光印刷厂
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
经　　销：各地新华书店
开　　本：787 × 980　1/16　　印张：35.25　字数：790千字
印　　次：2004年4月第1次印刷
定　　价：49.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

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

Preface

This book is intended as an introduction to object-oriented programming (OOP) using ANSI C++ for the reader or student who already has programming experience. It explains C++ features in the context of OOP.

C++ has had many recent additions including STL, namespaces, RTTI, and the bool type. These can be used readily by someone already proficient in basic C++, but most books have yet to treat these topics. This book can provide a handy guide to these new constructs.

The examples both within the book, and accessible at Addison-Wesley's web site are intended to exhibit good programming style. The Addison-Wesley web site, *www.aw.com* for this book contains the programs in the book as well as adjunct programs that illustrate points made in the book, or that flesh out short pieces of programs. The programs available at the web site are introduced by their *.cpp* or *.h* names and can be obtained by referencing

`www.aw.com/cseng/authors/pohl/opus2e/`*`program_name`*`.cpp`

C++, invented at Bell Labs by Bjarne Stroustrup in the mid-1980s, is a powerful modern successor language to C. C++ adds to C the concept of *class*, a mechanism for providing user-defined types also called *abstract data types*. It supports object-oriented programming by these means and by providing inheritance and run-time type binding.

By carefully developing working C++ programs, using the method of dissection, this book presents a simple and thorough introduction to the programming process in C++. Dissection is a technique for explaining new elements in a program that the student is seeing for the first time. It highlights key points in the many examples of working code that are used to teach by example.

This book is intended for use in a first course in programming in C++. It can be used as a supplementary text in an advanced programming course, data structures course, software methodology course, comparative language course, or other courses where the instructor wants C++ to be the language of choice. Each chapter presents a number of carefully explained programs. Many programs and functions are dissected.

All the major pieces of code were tested. A consistent and proper coding style is adopted from the beginning. The style standard used is one chosen by professionals in the C++ community.

In conjunction with *A Book on C, Third Edition* by Al Kelley and Ira Pohl (Addison Wesley Longman, 1995), an integrated treatment of the C and C++ programming languages and their use are presented which are not available elsewhere. For the beginner, a simpler introduction to the C language is *C by Dissection: The Essentials of C Programming, Third Edition* by Al Kelley and Ira Pohl (Addison Wesley Longman, 1995).

Chapters contain:

Object-Oriented Concept: Explains how an object-oriented programming concept is supported by a language feature.

Working Code: Small examples of working code illustrate concepts. Code illustrates a language feature or an OOP concept.

Dissections: A program particularly illustrative of the chapter's themes is analyzed by dissection. Dissection is similar to a structured walk-through of the code. Its intention is to explain to the reader newly encountered programming elements and idioms.

Pragmatics: Tips, pitfalls, nuances, and advice on the topic.

Summary: A succinct list of points are reiterated as helpful chapter review.

Exercises: The exercises test the student's knowledge of the language. Many exercises are intended to be done interactively while reading the text. This encourages self-paced instruction by the reader. The exercises also frequently extend the reader's knowledge to an advanced area of use.

The book incorporates:

Object-Oriented Programming. Object-Orientation is stressed throughout. Chapter 1, "Why Object-Oriented Programming in C++?," provides an introduction to C++'s use as an object-oriented programming language. Chapter 2, "Native Types and Statements," shows data types, expressions, and simple statements. Chapter 3, "Functions and Pointers," continues with similarities between functions and complex data types. The middle chapters show how classes work. Classes are the basis for abstract data types and object-oriented programming. The last few chapters give advanced details of the use of inheritance, templates, and exceptions. Chapter 12, "OOP Using C++," discusses OOP and the *Platonic* programming philosophy. This book develops in the programmer an appreciation of this point of view. At any point in the text the programmer can stop and use the new material.

Teaching by Example. This book is a tutorial that stresses examples of working code. From the start the student is introduced to full working programs. An interactive environment is assumed. Exercises are integrated with the examples to encourage experimentation. Excessive detail is avoided in explaining the larger elements of

writing working code. Each chapter has several important example programs. Major elements of these programs are explained by dissection.

Data Structures in C++. The text emphasizes many of the standard data structures from computer science. Stacks, safe arrays, dynamically allocated multidimensional arrays, lists, trees, and strings are all implemented. Exercises extend the student's understanding of how to implement and use these structures. Implementation is consistent with an abstract data type approach to software.

Standard Template Library (STL). STL is explained and used in Chapter 9, "Templates, Generic Programming, and STL." Many of the data structure examples foreshadow its explanation and use. There is a strong emphasis on the template mechanism required for STL and the iterator idiom that STL exploits.

ANSI C++ language and *iostream.h*. For an existing, widely used language, C++ continues to change at a rapid pace. This book is based on the most recent standard: the ANSI C++ Committee language documents. A succinct informal language reference is provided in Appendix C, "Language Guide." Chief additions include templates and exception handling. The examples use the *iostream.h* I/O library. This has replaced *stdio.h* used in the C community. Use of the *iostream.h* library is described in Appendix D, "Input/Output."

Reference Value in Appendices. There is an easily accessible informal language reference appendix: Appendix C, "Language Guide." Though this is not official, it specifies the language definition in a terse manner. There is also an appendix on the key I/O libraries, *iostream.h* and *stream.h*: Appendix D, "Input/Output." A short guide to both the *string* library and STL is given in Appendix E, "STL and String Libraries."

Idiomatic and Mainstream. The book attempts to stay with mainstream aspects of the language that are most important for the student and professional. It avoids arcane features of the language that are error prone or confusing. It is idiomatic in its use of code. The code is readily copied and reapplied to other problems.

Industry- and Course-Tested. It is the basis of many on-site professional training courses given by the author, who has used its contents to train professionals and students in various forums since 1986. The various changes in the new edition are course-tested, and reflect considerable teaching and consulting experience by the author. The book is the basis for an extensive series of video training tapes and on-line courses. More information on these courses is available at the author's web site at www.cse.ucsc.edu/~pohl.

Acknowledgments

My special thanks to my wife, Debra Dolsberry, who encouraged me throughout this project. She acted as book designer and technical editor for this second edition. She developed appropriate formats and style sheets in FrameMaker 4.0 and guided the transition process from the first edition in *troff*. She also implemented and tested all major pieces of code. Her careful implementations of the code and exercises led to many improvements. Stephen Clamage of TauMetric Corporation provided wonderfully insightful comments on language detail. William Engles of University of Wisconsin described an improved shuffling routine for the poker example. Reviews for this addition were provided by Jean Bell, Colorado School of Mines; Arthur Delcher, Loyola University; Konstantin Läufer, Loyola University; James L. Murphy, California State University; Kent Wooldridge, California State University; Shih-Ho Wang, University of California; David B. Teague, Western Carolina University; Lukasz Pruski, California State University; and David Gregory. Randal Burns and Hiroya Chiba, teaching assistants and computer science graduate students of University of California at Santa Cruz, also contributed to the reviewing process.

The first edition had help, inspiration, and encouragement from, Peter Apers, University of Twente, The Netherlands; Henri Bal, Vrije University, The Netherlands; Michael Beeson, State University of California; Nan Borreson, Borland International; Douglas Campbell, University of Conneticut; Cathy Collins, USC; Steve Demurjian; Robert Doran, University of Auckland, New Zealand; Robert Durling, UCSC; Daniel Edelson, UCSC; Anton Eliens, Vrije University, The Netherlands; Ray Fujioka, USC; Thomas Judson, University of Portland; Al Kelley, UCSC; Jim Kempf, Sun Microsystems, Incorporated; Darrell Long, UCSC; Charlie McDowell, UCSC; Laura Pohl, Cottage Consultants; Reind van de Riet, Vrije University, The Netherlands; Anthony Wasserman, IDE; and Salih Yurttas, Texas A&M University.

The second edition was developed with the support of my editor J. Carter Shanklin and editorial assistant Angela Buenning. Finally, I thank Bjarne Stroustrup for inventing such a powerful language and encouraging others to help teach it.

Ira Pohl
University of California, Santa Cruz

Contents