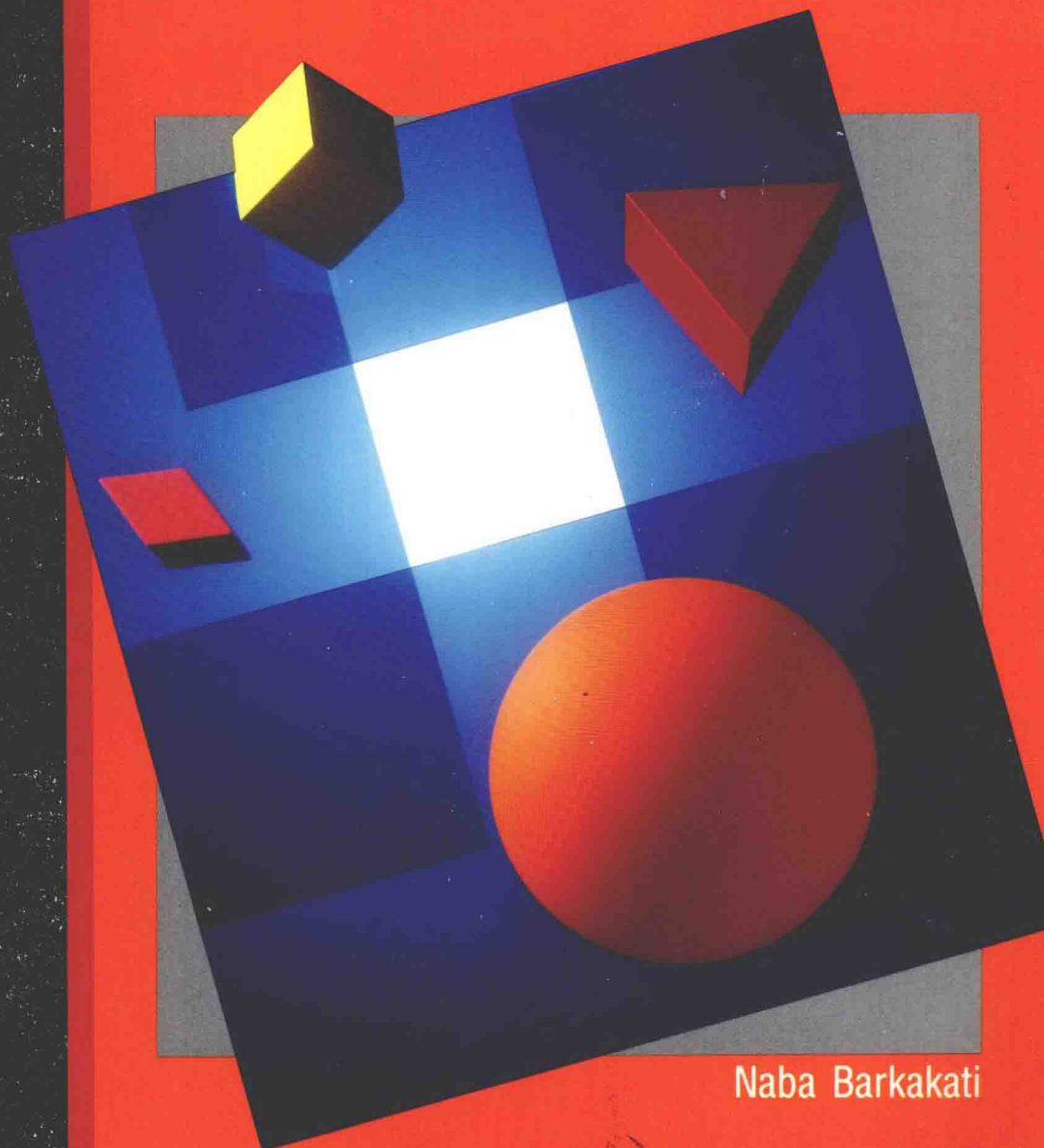


SAMS

OBJECT-ORIENTED PROGRAMMING IN C++

- Complete tutorial to object-oriented programming
- Detailed explanations of OOP software design principles
- Covers C++ Version 2.1



Naba Barkakati

Object-Oriented Programming in C++

Nabajyoti Barkakati

SAMS

*A Division of Prentice Hall Computer Publishing
11711 North College, Carmel, Indiana 46032 USA*

To Leba, Ivy, Emily, and Ashley

The following statement appears in the InterViews Release 2.6 documentation:

Copyright © 1987, 1988, 1989 Stanford University

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Stanford not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Stanford makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

STANFORD DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL STANFORD 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 THIS SOFTWARE.

FIRST EDITION

FOURTH PRINTING—1993

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. For information, address SAMS, 11711 N. College Ave., Carmel, IN 46032.

International Standard Book Number: 0-672-22800-9

Library of Congress Catalog Card Number: 91-60443

Publisher: *Richard K. Swadley*

Publishing Manager: *Joseph B. Wikert*

Managing Editor: *Neweleen Trebnik*

Senior Editor: *Rebecca Whitney*

Manuscript Editor: *Lynn Brown, Brown Editorial Service*

Production Editor: *Katherine Stuart Ewing*

Technical Editor: *Les Dye*

Cover Artist: *Dan Armstrong*

Production Assistance: *Jeff Baker, Brad Chinn, Martin Coleman,*

Sandy Grieshop, Dennis Clay Hager, Louise Shinault

Indexers: *Jill Bomaster, Johnna VanHoose*

Printed in the United States of America

Object-Oriented Programming in C++

Preface

Slowly but inexorably, object-oriented programming (OOP) is becoming the method of choice for software design, and with it, C++, the language of choice among programmers. Loosely speaking, OOP is a new way of organizing software that is based on real-world objects. Although OOP techniques can be applied in any language (C, for instance), a language such as C++ that is designed to support OOP makes it easier to implement the OOP techniques. Although the use of OOP does not impart anything to a finished software product that the user can see, the software developer can gain substantial advantages by using OOP, especially in large software projects. Because OOP allows the software to remain close to the conceptual, higher-level model of the real-world problem, the programmer can manage the complexity better than an approach that requires mapping the problem to fit the features of the language.

C++ is a superset of C with features designed to support OOP. C++ was developed in the early 1980s by Bjarne Stroustrup of AT&T Bell Laboratories. He created C++ while he was enhancing C to support efficient event-driven simulation programs. His inspiration came from the language Simula67, which supported the concept of a *class*. AT&T made many improvements to this initial language before releasing it commercially for the first time in 1985. Since then, C++ has continued to evolve, with AT&T controlling the releases.

In the beginning, AT&T supplied a translator called *cfront* for converting C++ programs into C, which were then compiled using a C compiler. By the time Release 1.2 of AT&T's C++ was available, C++ compilers such as the one from Zortech, Inc., were becoming available for PCs and workstations. AT&T released C++ 2.0 in 1989 and followed it promptly with Release 2.1, a maintenance release. Today, C++ translators and compilers are available on a wide variety of systems, including MS-DOS PCs, Apple Macintoshes, and all UNIX systems.

A good indication of the popularity of C++ and OOP is the large number of books published on these two topics. However, most of the books suffer from one or more drawbacks. Those that cover object-oriented design principles do not pay much attention to C++. The ones that are designed to teach C++ do not adequately describe OOP. Up to now, programmers have had to go through at least two or three books to learn OOP and C++. There is a definite need for a single source of information for these two topics.

Object-Oriented Programming in C++ is designed to answer the needs of C programmers learning C++ and OOP. This book teaches OOP and C++ and, at the same time, serves as a reference guide to the C++ programming language. It includes in-depth tutorials that gently introduce you to OOP and show you how to apply these techniques in C++.

The book features

- extensive tutorials on the basic concepts of object-oriented programming.
- discussion of data abstraction, inheritance, and polymorphism.
- a step-by-step introduction to the features of C++ such as classes and virtual functions that support OOP.
- complete coverage of the latest release of AT&T's C++.
- discussions of how C++ differs from ANSI standard C.
- detailed examples showing how to use OOP techniques.
- descriptions of the different approaches to building a library of reusable classes in C++.
- coverage of object-oriented software design principles.
- real-world examples of using C++ class libraries in applications meant for Microsoft Windows 3.0 and the X Window System.
- a glossary of OOP terminology.
- A minireference to the C++ programming language, especially its keywords.

Instead of going through a litany of syntactical details, *Object-Oriented Programming in C++* uses many short sample programs to illustrate OOP techniques. Features of C++ are always presented in the context of an OOP concept that the feature supports.

Although the descriptions of OOP and C++ rely on generic AT&T C++, *Object-Oriented Programming in C++* also includes coverage of commercial class libraries that are useful for building real-world applications. Specifically, this book describes M++, C++/Views, and InterViews toolkits—sets of classes that you can use in your applications. Additionally, you will find examples of writing Microsoft Windows programs in C++.

It is easy to get overwhelmed by the new syntax of C++ and the details of how everything fits together in a program that uses an object-oriented design. However, with a grasp of the fundamentals of OOP and with the help of C++ class libraries, you will find it relatively easy to employ OOP techniques in your applications. I sincerely hope that *Object-Oriented Programming in C++* will get you started on your way to harnessing the full power of object-oriented techniques and C++.

Acknowledgments

I am grateful to Joe Wikert for suggesting the idea of this book—a single source of information for C programmers interested in learning both C++ and object-oriented programming. Thanks to Joe for not only suggesting the idea but also for getting me started on this project and seeing it through to its successful completion.

Thanks to Nan Borreson of Borland International for providing me the beta copies of the Borland C++ 2.0 compiler. Thanks also to Bob Swarm of Zortech for sending me a copy of Zortech C++ Version 2.1. Barbara LoFranco of The Santa Cruz Operations was kind enough to send me a copy of SCO C++ as soon as it was released—thus making it possible to write Chapter 15 and test some of the examples under SCO C++. The folks at SCO, especially Barbara LoFranco and Allen Ginzburg, helped me immensely by providing copies of Open Desktop Personal System and Open Desktop Development System so that I could use the X Window System, Motif, and SCO C++.

For the chapter on C++ class libraries, I needed some sample libraries. Les Dye of Dyad Software and Jim Schwarz of CNS came through with *M++* and *C++/Views*, respectively. Thanks to both of you for helping me out. Les Dye deserves a second round of thanks for the technical review of the manuscript.

Production of a book like this always involves many dedicated professionals doing their part behind the scenes. I would like to give my heartfelt thanks to each and every one involved in turning my raw manuscript into this well-edited, beautifully packaged book. In particular, thanks to Lynn Brown of Brown Editorial Service for the thorough editing of the manuscript and to Kathy Ewing for managing the production.

Of course, there would be no reason for this book if it were not for the C++ programming language. For this, I have Bjarne Stroustrup, the principal author of C++, to thank.

Finally, my greatest thanks go to my wife Leha for her patience and understanding and for taking care of everything while I stayed glued to my PCs. I can't imagine how she managed, because our daughter Ashley was born midway through this book project. As I wrap up the book, Ashley is already hard at work encouraging me with her smiles, while her older sisters Ivy and Emily are tracking my progress and counting the days to the deadline. Thanks for being there!

Nabajyoti Barkakati

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks are listed below. In addition, terms suspected of being trademarks or service marks have been appropriately capitalized. SAMS cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Ada is a registered trademark of the U.S. Department of Defense.

Apple, MacApp, and Macintosh are registered trademarks and AppleTalk is a trademark of Apple Computer, Inc.

AT&T is a registered trademark of American Telephone and Telegraph Company.

BSD is a trademark of University of California, Berkeley.

Borland and Borland C++ are registered trademarks and Turbo Assembler, Turbo C++, and Turbo Profiler are trademarks of Borland International, Inc.

C++/Views is a trademark of CNS, Inc.

CompuServe is a registered trademark of CompuServe, Inc.

dbXtra is a trademark of SCO Canada, Inc.

DEC is a registered trademark and DECnet, Ultrix, VAX, VAXstation II/GPX, VAX/VMS, and VMS are trademarks of Digital Equipment Corporation.

Dyad and M++ are trademarks of Dyad Software Corporation.

Eiffel is a registered trademark of Interactive Software Engineering, Inc.

Ethernet is a trademark of Xerox Corporation.

Helvetica and Times are trademarks of Linotype Company.

Hewlett-Packard and HP are registered trademarks of Hewlett-Packard Company.

IBM, IBM PC, IBM XT, IBM AT are registered trademarks and OS/2 and Presentation Manager are trademarks of International Business Machines.

Intel is a registered trademark and Intel 8088, 8086, 80186, 80286, 80386, 80486, i860, 8087, 80287, and 80387 are trademarks of Intel Corporation.

Microsoft, MS-DOS, and XENIX are registered trademarks of Microsoft Corporation.

Motif, OSF, and OSF/Motif are trademarks of Open Software Foundation, Inc.

Motorola MC68000 is a trademark of Motorola, Inc.

Objective-C and Software-IC are registered trademarks of The Stepstone Corporation.

Open Desktop and SCO are registered trademarks and the Santa Cruz Operation is a trademark of The Santa Cruz Operation, Inc.

OPEN LOOK and System V are trademarks of AT&T.

PostScript is a registered trademark of Adobe Systems, Inc.

Simula67 is a trademark of Simula AS.

Smalltalk-80 is a trademark of ParcPlace Systems.

Software Through Pictures is a registered trademark of Interactive Development Environments, Inc.

SPARCstation, Sun, Sun-3, Sun-4, Sun Microsystems, Sun Workstation, and SunOS are registered trademarks of Sun Microsystems, Inc.

TEKTRONIX is a registered trademark of Tektronix, Inc.

Times Roman is a trademark of Monotype Corporation.

UNIX is a registered trademark of UNIX System Laboratories, Inc.

Whitewater Resource Toolkit is a trademark of The Whitewater Group.

X Window System is a trademark of The Massachusetts Institute of Technology.

Zinc and Zinc Interface Library are trademarks of Zinc Software, Inc.

Zortech is a trademark of Zortech, Inc.

Introduction

O*bject-Oriented Programming in C++* is an intermediate-level book that introduces you to the basic concepts of object-oriented programming (OOP) and shows you how to apply OOP techniques using the C++ programming language. This book assumes that you already know the C programming language. The goal is to assist you, the C programmer, in becoming familiar with the terminology of OOP and to show you how various features of C++ support OOP.

To this end, *Object-Oriented Programming in C++* focuses on the basic concepts of OOP, how OOP helps you handle changes in software requirements easily, and how C++ supports OOP. This book also covers programming with the *iostream* I/O library and shows how to call C functions from C++, organize C++ class libraries, and use commercially available C++ class libraries. Once you have mastered the basics of OOP, C++, and C++ class libraries, *Object-Oriented Programming in C++* moves on to the subject of developing realistic applications in C++. The book includes sample programs for MS-DOS, Microsoft Windows, and the X Window System. Figures are used to illustrate concepts and show the inheritance hierarchies of classes.

Although *Object-Oriented Programming in C++* includes a reference section on the C++ programming language, it is *not* a complete reference for C++. This book's goal is to show you how to use C++ to create and use objects, not simply to present the language definition for compiler writers.

What You Need

To make the best use of this book, you should have access to a system with a C++ compiler. That way, you can test the example programs as you progress through the book. For those using MS-DOS PCs, the latest versions of Borland C++ or Zortech C++ fits the bill perfectly. On Intel 80386 PCs running UNIX System V, The Santa Cruz Operation's SCO C++ is a possible choice. For most other UNIX systems, C++ compilers are available directly from the system's vendor. If your system does not have a C++ compiler, you may be able to find a copy of Free Software Foundation's GNU C++ (g++) that works on your system.

Of course, if you want to use C++ to develop applications for Microsoft Windows or the X Window System, you will need some additional items. Chapters 14 and 15 describe what you need to use C++ for Microsoft Windows and X applications, respectively.

All examples in this book were tested with Borland C++ 2.0 on two systems: a 1985 vintage 6-MHz IBM PC-AT and an Intel 80386-based ISA (Industry Standard

Architecture) PC with 8MB of memory, a 150MB hard disk, and a VGA display. Many examples were also tested using Zortech C++ version 2.1. (I could not use Zortech C++ for many examples, because Zortech C++ 2.1 lacks the *iostream* library required by the examples.) When needed, I ran SCO Open Desktop (which includes UNIX System V/386 Release 3.2, X version 11 Release 3, and OSF/Motif version 1.0) on the 80386 system. All the examples should compile and link on most systems without any change. They do require the *iostream* library, however.

Conventions Used in This Book

Object-Oriented Programming in C++ uses a simple notational style. All listings are typeset in a monospace font for ease of reading. All file names, function names, variable names, and keywords appearing in text are also in the same monospace font. The first occurrence of new terms and concepts is in *italic*. Notes, which appear in boxes, explain terms and concepts that appear in the text nearby.

How to Use This Book

If you are a newcomer to C++, you should read the book from front to back. The sample programs usually build on classes introduced in earlier chapters. For example, a sample program in Chapter 13 might need classes that are presented in Chapter 8. If you read the chapters in order, you will be aware of this and be able to understand the examples easily.

There are five parts in the book. The first four parts comprise 16 chapters. Part I includes three chapters that explain the basic concepts of object-oriented programming (OOP). Part II has six chapters that describe how to use the features of C++ that support OOP. Part III includes three chapters that show how to organize C++ class libraries and how to use commercially available libraries. Part IV consists of four chapters that present realistic applications for MS-DOS and Microsoft Windows. Part IV also describes the InterViews toolkit that you can use to develop applications for the X Window System. The final part, Part V, is a collection of four appendixes that include a glossary and a list of C++ compilers and libraries, as well as the comprehensive index to the book.

From this quick overview, you can decide whether you want to skip any of the parts. For example, if you are already familiar with the basic terminology of OOP, you can skip Part I and go straight to Part II. On the other hand, if you know how C++ supports OOP and want to start using C++ in realistic applications, you can skip Parts I through III and start with Chapter 13. To help you decide how to best use the book, the following sections describe the chapters in greater detail.

Part I: Learning Object-Oriented Programming

This part of the book introduces you, the prospective C++ programmer, to the basic concepts of object-oriented programming: data abstraction, inheritance, and polymorphism. Examples in C illustrate how you can apply these concepts in practice. You will see the same examples in C++ and see how an object-oriented language simplifies the use of OOP techniques. At this point you do not have to know C++, but you will be asked to go along with the assumption that everything will be explained in later chapters. The final chapter in Part I describes the process of designing software the “object-oriented” way. Part I includes

- Chapter 1: Basics of Object-Oriented Programming
- Chapter 2: C++ and Object-Oriented Programming
- Chapter 3: Object-Oriented Software Design

Part II: Learning C++

Part II teaches the C++ programming language from the perspective of OOP. Chapter 4 provides a brief review of ANSI standard C and a summary of the new features of C++. This chapter serves as a refresher on C and gives a complete overview of C++. Chapter 5 shows how to use the predefined stream classes in the `iostream` library to perform I/O. The next four chapters explain how various facilities of C++ are used to define, create, and manipulate objects. These chapters show the use of C++ features such as classes, inheritance, and virtual functions. Part II also covers overloading of functions and operators. The chapters in Part II include

- Chapter 4: C++ and ANSI Standard C
- Chapter 5: Predefined Classes in C++
- Chapter 6: Building Objects with Classes
- Chapter 7: Defining Operations on Objects
- Chapter 8: Using Inheritance in C++
- Chapter 9: Virtual Functions and Polymorphism

Part III: Applying OOP Techniques in C++

Part III introduces you to the idea of building and using libraries of C++ classes. This part explains how to use C libraries in C++ programs. Chapter 11 describes some strategies for building C++ class libraries. The last chapter in this part summarizes the capabilities of two representative commercial class libraries and discusses how the programmer can supplement in-house libraries with these commercial offerings. Part III includes the following chapters:

Chapter 10: Using C Libraries in C++ Programs

Chapter 11: Building Class Libraries in C++

Chapter 12: Using Commercial Class Libraries

Part IV: Developing Applications in C++

Part IV is devoted to illustrating how to use C++ class libraries in realistic applications. Chapter 13 starts with the basic building blocks for a Forms package—software that you use to create, store, and display forms. Because the user interface benefits the most from an OOP approach, the next chapter focuses on building the user interface for the Forms software under Microsoft Windows. Chapter 15 describes the InterViews toolkit which you can use to create applications that use the X Window System. Chapter 16 covers a number of advanced topics, such as exception-handling and templates that are being considered for addition to the C++ programming language. Part IV includes the following chapters:

Chapter 13: Building MS-DOS Applications in C++

Chapter 14: Developing for Microsoft Windows in C++

Chapter 15: C++ and the X Window System

Chapter 16: Advanced Topics in C++

Part V: Appendixes and Index

Part V starts with a glossary of terms used in OOP. This is followed by Appendix B which serves as a minireference for the C++ programming language. It includes reference entries for the keywords. Appendix C is a list of currently available C++ compilers and class libraries. Appendix C also includes information on how to obtain these libraries. Appendix D provides a summary of the standard ANSI C library. You will find the prototypes of all standard C library routines here.

Appendix A: Glossary

Appendix B: C++ Language Reference

Appendix C: C++ Compilers and Class Libraries

Appendix D: ANSI C Headers

Index

How to Contact the Author

If you have any questions or suggestions, or if you want to report any errors, please feel free to contact me either by mail or through electronic mail. Here is how:

- Write to LNB Software, Inc., 2005 Aventurine Way, Silver Spring, MD 20904
- If you have access to an Internet node, send E-mail to:
naba@grebyn.com
- If you use CompuServe, specify the following as SEND TO:
>INTERNET:naba@grebyn.com
- From MCIMAIL, specify the following when sending mail:
EMS: INTERNET
MBX: naba@grebyn.com

Please do not phone, even if you happen to come across my telephone number. Instead, drop me a letter or send an E-mail message, and you are guaranteed a prompt reply.

How to Get the Source Code on Disk

For your convenience, the complete source code for the programs in *Object-Oriented Programming in C++* is available on MS-DOS disks. Use the order form at the back of the book to request your copy. If the form is missing, send \$30 (money order or check in U.S. funds drawn on a U.S. bank) to

LNB Software, Inc.
2005 Aventurine Way
Silver Spring, MD 20904

Foreign orders, please add \$6 for shipping and handling. Maryland residents, please add sales tax. Please specify diskette size (5¼ inch or 3½ inch).

Overview

Preface, xxi
Acknowledgments, xxiii
Introduction, xxv

Part I Learning Object-Oriented Programming, 1

- 1** Basics of Object-Oriented Programming, 3
- 2** C++ and Object-Oriented Programming, 39
- 3** Object-Oriented Software Design, 61

Part II Learning C++, 87

- 4** C++ and ANSI Standard C, 89
- 5** Predefined Classes in C++, 137
- 6** Building Objects with Classes, 161
- 7** Defining Operations on Objects, 189
- 8** Using Inheritance in C++, 231
- 9** Virtual Functions and Polymorphism, 277

Part III Applying OOP Techniques in C++, 293

- 10** Using C Libraries in C++ Programs, 295
- 11** Building Class Libraries in C++, 329
- 12** Using Commercial Class Libraries, 345

Part IV Developing Applications in C++, 375

- 13** Building MS-DOS Applications in C++, 377
- 14** Developing Programs for Microsoft Windows in C++, 459
- 15** C++ and the X Window System, 535
- 16** Advanced Topics in C++, 575

Part V Appendixes and Index, 593

- A** Glossary, 595
- B** C++ Language Reference, 599
- C** C++ Compilers and Class Libraries, 625
- D** ANSI C Headers, 633
- Index, 653

Contents

Preface, xxi
Acknowledgments, xxiii
Introduction, xxv

PART I Learning Object-Oriented Programming, 1

1 Basics of Object-Oriented Programming, 3

What Is Object-Oriented Programming?, 4
 Procedure-Oriented Programming, 5
 An Example in C, 6
 Adding a New Shape, 11
OOP Terminology, 13
 Data Abstraction, 14
 Objects, Classes, and Methods, 14
Inheritance, 15
 Multiple Inheritance, 16
Polymorphism, 16
OOP in C, 17
 Defining Objects in C, 17
 Implementing Geometrical Shapes, 21
 Allocating an Object's Data, 25
 Utility Functions, 29
 Using the Shapes, 31
 Adding a New Shape Object, 32
 Problems with OOP in C, 36
Summary, 37
Further Reading, 37

2 C++ and Object-Oriented Programming, 39

Object-Oriented Programming in C++, 40
 Data Abstraction in C++, 41
 Defining the Member Functions for File, 43
 Constructors and Destructors, 45
 Using the File Class, 46

- Inheritance in C++ Classes, 46
- Polymorphism and Dynamic Binding, 47
- Geometric Shapes in C++, 49
 - The Shape Classes, 49
 - Circle and Rectangle Classes, 51
 - Using the Shape Classes, 53
 - Adding a New Shape Class, 54
 - Creating Objects at Run-Time, 56
- Summary, 57
- Further Reading, 57

3 Object-Oriented Software Design, 61

- Software Engineering Trends, 61
 - Prevailing Methods, 63
 - Problems with the Top-Down Approach, 65
- Object-Oriented Approach, 65
 - Finding Objects Using Structured Analysis Tools, 66
 - Notation for Objects, 69
 - Hybrid Design Methods, 69
 - Using Physical Metaphors: The Desktop, 71
 - Separation of Responsibilities, 73
 - Model-View-Controller (MVC) Architecture of Smalltalk-80, 74
 - Influence of the MVC Architecture, 75
 - Applying the Principle of Separation, 76
 - Responsibility-Driven Design, 78
 - Design Phases, 78
 - The Design Tools, 79
 - Step-by-Step Design, 82
- Summary, 82
- Further Reading, 83

Part II Learning C++, 87

4 C++ and ANSI Standard C, 89

- Overview of ANSI Standard C, 89
 - Structure of a C Program, 90
 - ANSI C Escape Sequences and Trigraphs, 93
 - Preprocessor Directives, 94
 - Including Files, 95
 - Defining Macros, 96
 - Conditional Directives, 98
 - Other Directives, 99

Declaration and Definition of Variables, 99	
Basic Types, 100	
Enumerations, 101	
Structures, Unions, and Bit-fields, 101	
Arrays, 103	
Pointers, 104	
Type Definitions, 105	
Type Qualifiers: const and volatile, 105	
Expressions, 106	
Operator Precedence, 108	
Statements, 110	
break Statement, 111	
case Label, 111	
Compound Statement or Block, 111	
continue Statement, 112	
default Label, 112	
do Statement, 112	
Expression Statements, 113	
for Statement, 113	
goto Statement, 113	
if Statement, 114	
if . . else Statement, 114	
Null Statement, 115	
return Statement, 115	
switch Statement, 115	
while Statement, 116	
Functions, 117	
Function Prototypes, 117	
The void Type, 118	
Functions with Varying Numbers of Arguments, 118	
The ANSI C Library, 119	
Features of C++, 120	
New Features for Functions in C++, 120	
Default Arguments, 121	
Overloaded Function Names, 122	
Inline Functions, 122	
friend Functions, 124	
Reference Types as Arguments, 125	
Overloaded Operators, 128	
Data Declarations in C++, 129	
How C++ Differs from C, 130	
New Reserved Keywords, 130	
Function Prototypes, 131	
const Variables, 131	
void Pointers, 132	
Initialization of Character Arrays, 133	
