

OSBORNE



International Best-Seller!

Over 200,000 Copies Sold!

# TEACH YOURSELF

"Herb Schildt tells his programmers what they want and need to know—simply, clearly, concisely, and authoritatively."  
-ACM Computing Reviews

NEW EDITION  
2001



# C++

Third  
Edition

The Most Successful and Proven Method for Learning C++

- Master Standard C++ - Including its new features!
- Essential for C Programmers moving on to C++
- Build your skills with hundreds of examples and exercises
- Covers the Standard Template Library (STL)

## Herbert Schildt

Best-Selling C/C++ Author with More Than 1.5 Million Books Sold

*Teach*

---

*Yourself C++,*

---

*Third Edition*

**Herbert Schildt**

Osborne **McGraw-Hill**

Berkeley New York St. Louis San Francisco Auckland Bogotá Hamburg London Madrid Mexico City  
Milan Montreal New Delhi Panama City Paris São Paulo Singapore Sydney Tokyo Toronto

**PUBLISHER** Osborne/McGraw-Hill  
*Brandon A. Nordin* 2600 Tenth Street  
Berkeley, California 94710  
U.S.A.

**EDITOR-IN-CHIEF**  
*Scott Rogers*

**ACQUISITIONS EDITOR** For information on translations or book distributors outside the U.S.A., or  
*Megg Bonar* to arrange bulk purchase discounts for sales promotions, premiums, or  
fund-raisers, please contact Osborne/McGraw-Hill at the above address.

**PROJECT EDITOR**  
*Janet Walden*

Teach Yourself C++, Third Edition

**EDITORIAL ASSISTANT**  
*Gordon Hurd*

**TECHNICAL EDITOR**  
*Greg Guntle*

Copyright © 2000 by The McGraw-Hill Companies. All rights reserved.  
Printed in the United States of America. Except as permitted under the  
Copyright Act of 1976, no part of this publication may be reproduced or  
distributed in any form or by any means, or stored in a database or retrieval  
system, without the prior written permission of the publisher, with the  
exception that the program listings may be entered, stored, and executed in a  
computer system, but they may not be reproduced for publication.

**COPY EDITOR**  
*Katherine Krause*

**Reprint 2001**

1234567890 DOC DOC 901987654321098

**PROOFREADERS**  
*Pat Mannion*  
*Emily Wolman*

ISBN 0-07-882392-7

**INDEXER**  
*Sheryl Schildt*

**COMPUTER DESIGNER**  
*Jani Beckwith*

Information has been obtained by Osborne/McGraw-Hill from sources  
believed to be reliable. However, because of the possibility of human or  
mechanical error by our sources, Osborne/McGraw-Hill, or others,  
Osborne/McGraw-Hill does not guarantee the accuracy, adequacy, or  
completeness of any information and is not responsible for any errors or  
omissions or the results obtained from use of such information.

**ILLUSTRATOR**  
*Roberta Steele*

**SERIES DESIGN**  
*Roberta Steele*

**COVER DESIGN**  
*Arlette Crosland*

# Contents at a Glance

1	<i>An Overview of C++</i>	1
2	<i>Introducing Classes</i>	41
3	<i>A Closer Look at Classes</i>	87
4	<i>Arrays, Pointers, and References</i>	117
5	<i>Function Overloading</i>	159
6	<i>Introducing Operator Overloading</i>	195
7	<i>Inheritance</i>	231
8	<i>Introducing the C++ I/O System</i>	269
9	<i>Advanced C++ I/O</i>	307
10	<i>Virtual Functions</i>	345
11	<i>Templates and Exception Handling</i>	371
12	<i>Run-Time Type Identification and the Casting Operators</i>	407
13	<i>Namespaces, Conversion Functions, and Miscellaneous Topics</i>	435
14	<i>Introducing the Standard Template Library</i>	473
A	<i>A Few More Differences Between C and C++</i>	531
B	<i>Answers</i>	533
	<i>Index</i>	739

---

## **A** *about the Author...*

Herbert Schildt is the world's leading programming author. He is an authority on the C and C++ languages, a master Windows programmer, and an expert on Java. His programming books have sold nearly two million copies worldwide and have been translated into all major foreign languages. He is the author of numerous best-sellers, including *C: The Complete Reference*, *C++: The Complete Reference*, *C++ from the Ground Up*, *Expert C++*, *MFC Programming from the Ground Up*, *Windows 95 Programming in C and C++*, *Windows NT 4 Programming from the Ground Up*, and many others. Schildt is the president of Universal Computing Laboratories, a software consulting firm in Mahomet, Illinois. He is also a member of both the ANSI C and C++ standardization committees. He holds a master's degree in computer science from the University of Illinois.

# Contents

*Acknowledgments*, xiii

*Introduction*, xv

*For Further Study*, xix

## **1 An Overview of C++ . . . 1**

- 1.1 WHAT IS OBJECT-ORIENTED PROGRAMMING?, 3
- 1.2 TWO VERSIONS OF C++, 7
- 1.3 C++ CONSOLE I/O, 13
- 1.4 C++ COMMENTS, 19
- 1.5 CLASSES: A FIRST LOOK, 21
- 1.6 SOME DIFFERENCES BETWEEN C AND C++, 28
- 1.7 INTRODUCING FUNCTION OVERLOADING, 33
- 1.8 C++ KEYWORDS, 39
- SKILLS CHECK, 39

## **2 Introducing Classes . . . 41**

- 2.1 CONSTRUCTOR AND DESTRUCTOR FUNCTIONS, 43
- 2.2 CONSTRUCTORS THAT TAKE PARAMETERS, 52
- 2.3 INTRODUCING INHERITANCE, 59
- 2.4 OBJECT POINTERS, 66
- 2.5 CLASSES, STRUCTURES, AND UNIONS ARE RELATED, 68
- 2.6 IN-LINE FUNCTIONS, 75
- 2.7 AUTOMATIC IN-LINING, 80
- SKILLS CHECK, 83

## **3 A Closer Look at Classes . . . 87**

- 3.1 ASSIGNING OBJECTS, 89
- 3.2 PASSING OBJECTS TO FUNCTIONS, 96

- 3.3 RETURNING OBJECTS FROM FUNCTIONS, 102
- 3.4 AN INTRODUCTION TO FRIEND  
FUNCTIONS, 107  
SKILLS CHECK, 114
  
- 4 Arrays, Pointers, and References . . . 117**
  - 4.1 ARRAYS OF OBJECTS, 119
  - 4.2 USING POINTERS TO OBJECTS, 124
  - 4.3 THE `this` POINTER, 126
  - 4.4 USING `new` AND `delete`, 130
  - 4.5 MORE ABOUT `new` AND `delete`, 134
  - 4.6 REFERENCES, 140
  - 4.7 PASSING REFERENCES TO OBJECTS, 146
  - 4.8 RETURNING REFERENCES, 149
  - 4.9 INDEPENDENT REFERENCES AND  
RESTRICTIONS, 154  
SKILLS CHECK, 156
  
- 5 Function Overloading . . . 159**
  - 5.1 OVERLOADING CONSTRUCTOR  
FUNCTIONS, 161
  - 5.2 CREATING AND USING A COPY  
CONSTRUCTOR, 167
  - 5.3 THE `overload` ANACHRONISM, 177
  - 5.4 USING DEFAULT ARGUMENTS, 177
  - 5.5 OVERLOADING AND AMBIGUITY, 185
  - 5.6 FINDING THE ADDRESS OF AN OVERLOADED  
FUNCTION, 189  
SKILLS CHECK, 191
  
- 6 Introducing Operator Overloading . . . 195**
  - 6.1 THE BASICS OF OPERATOR OVERLOADING,  
197
  - 6.2 OVERLOADING BINARY OPERATORS, 199
  - 6.3 OVERLOADING THE RELATIONAL AND  
LOGICAL OPERATORS, 207
  - 6.4 OVERLOADING A UNARY OPERATOR, 209
  - 6.5 USING FRIEND OPERATOR FUNCTIONS, 213
  - 6.6 A CLOSER LOOK AT THE ASSIGNMENT  
OPERATOR, 218
  - 6.7 OVERLOADING THE `[]` SUBSCRIPT  
OPERATOR, 222  
SKILLS CHECK, 227

## **7 Inheritance . . . 231**

- 7.1 BASE CLASS ACCESS CONTROL, 234
- 7.2 USING PROTECTED MEMBERS, 240
- 7.3 CONSTRUCTORS, DESTRUCTORS, AND INHERITANCE, 244
- 7.4 MULTIPLE INHERITANCE, 252
- 7.5 VIRTUAL BASE CLASSES, 259
- SKILLS CHECK, 262

## **8 Introducing the C++ I/O System . . . 269**

- 8.1 SOME C++ I/O BASICS, 273
- 8.2 FORMATTED I/O, 275
- 8.3 USING width(), precision(), AND fill(), 283
- 8.4 USING I/O MANIPULATORS, 287
- 8.5 CREATING YOUR OWN INSERTERS, 292
- 8.6 CREATING EXTRACTORS, 299
- SKILLS CHECK, 303

## **9 Advanced C++ I/O . . . 307**

- 9.1 CREATING YOUR OWN MANIPULATORS, 309
- 9.2 FILE I/O BASICS, 313
- 9.3 UNFORMATTED, BINARY I/O, 320
- 9.4 MORE UNFORMATTED I/O FUNCTIONS, 327
- 9.5 RANDOM ACCESS, 331
- 9.6 CHECKING THE I/O STATUS, 334
- 9.7 CUSTOMIZED I/O AND FILES, 338
- SKILLS CHECK, 341

## **10 Virtual Functions . . . 345**

- 10.1 POINTERS TO DERIVED CLASSES, 347
- 10.2 INTRODUCTION TO VIRTUAL FUNCTIONS, 349
- 10.3 MORE ABOUT VIRTUAL FUNCTIONS, 357
- 10.4 APPLYING POLYMORPHISM, 362
- SKILLS CHECK, 368

## **11 Templates and Exception Handling . . . 371**

- 11.1 GENERIC FUNCTIONS, 373
- 11.2 GENERIC CLASSES, 380
- 11.3 EXCEPTION HANDLING, 386
- 11.4 MORE ABOUT EXCEPTION HANDLING, 394



11.5 HANDLING EXCEPTIONS THROWN

BY new, 401

SKILLS CHECK, 404

**12 Run-Time Type Identification and the Casting**

**Operators . . . 407**

12.1 UNDERSTANDING RUN-TIME TYPE

IDENTIFICATION (RTTI), 409

12.2 USING dynamic\_cast, 420

12.3 USING const\_cast, reinterpret\_cast, AND

static\_cast, 429

SKILLS CHECK, 432

**13 Namespaces, Conversion Functions, and**

**Miscellaneous Topics . . . 435**

13.1 NAMESPACES, 437

13.2 CREATING A CONVERSION FUNCTION, 446

13.3 STATIC CLASS MEMBERS, 449

13.4 const MEMBER FUNCTIONS AND mutable, 455

13.5 A FINAL LOOK AT CONSTRUCTORS, 459

13.6 USING LINKAGE SPECIFIERS AND THE asm  
KEYWORD, 463

13.7 ARRAY-BASED I/O, 466

SKILLS CHECK, 471

**14 Introducing the Standard Template Library . . . 473**

14.1 AN OVERVIEW OF THE STANDARD TEMPLATE  
LIBRARY, 476

14.2 THE CONTAINER CLASSES, 479

14.3 VECTORS, 480

14.4 LISTS, 490

14.5 MAPS, 502

14.6 ALGORITHMS, 509

14.7 THE string CLASS, 519

SKILLS CHECK, 529

**A A Few More Differences Between C and C++ . . . 531**

**B Answers . . . 533**

1.3 EXERCISES, 534

1.4 EXERCISES, 535

1.5	EXERCISES, 535	
1.6	EXERCISES, 538	
1.7	EXERCISES, 538	
	MASTERY SKILLS CHECK: Chapter 1, 541	
	REVIEW SKILLS CHECK: Chapter 2, 543	
2.1	EXERCISES, 545	
2.2	EXERCISES, 548	
2.3	EXERCISE, 551	
2.5	EXERCISES, 553	
2.6	EXERCISES, 555	
2.7	EXERCISES, 556	
	MASTERY SKILLS CHECK: Chapter 2, 558	
	CUMULATIVE SKILLS CHECK: Chapter 2, 560	
	REVIEW SKILLS CHECK: Chapter 3, 562	
3.1	EXERCISES, 563	
3.2	EXERCISES, 565	
3.3	EXERCISES, 567	
3.4	EXERCISE, 567	
	MASTERY SKILLS CHECK: Chapter 3, 569	
	CUMULATIVE SKILLS CHECK: Chapter 3, 571	
	REVIEW SKILLS CHECK: Chapter 4, 576	
4.1	EXERCISES, 578	
4.2	EXERCISES, 580	
4.3	EXERCISE, 582	
4.4	EXERCISES, 583	
4.5	EXERCISES, 584	
4.6	EXERCISES, 585	
4.7	EXERCISE, 586	
4.8	EXERCISES, 587	
	MASTERY SKILLS CHECK: CHAPTER 4, 589	
	CUMULATIVE SKILLS CHECK: Chapter 4, 592	
	REVIEW SKILLS CHECK: Chapter 5, 593	
5.1	EXERCISES, 595	
5.2	EXERCISES, 598	
5.4	EXERCISES, 600	
5.6	EXERCISE, 601	
	MASTERY SKILLS CHECK: Chapter 5, 602	
	CUMULATIVE SKILLS CHECK: Chapter 5, 605	
	REVIEW SKILLS CHECK: Chapter 6, 607	

- 6.2 EXERCISES, 608
- 6.3 EXERCISE, 609
- 6.4 EXERCISES, 610
- 6.5 EXERCISES, 612
- 6.6 EXERCISE, 616
- 6.7 EXERCISES, 618
  - MASTERY SKILLS CHECK: Chapter 6, 621
  - CUMULATIVE SKILLS CHECK: Chapter 6, 629
  - REVIEW SKILLS CHECK: Chapter 7, 631
- 7.1 EXERCISES, 637
- 7.2 EXERCISES, 637
- 7.3 EXERCISES, 638
- 7.4 EXERCISES, 640
- 7.5 EXERCISES, 641
  - MASTERY SKILLS CHECK: Chapter 7, 641
  - CUMULATIVE SKILLS CHECK: Chapter 7, 643
  - REVIEW SKILLS CHECK: Chapter 8, 644
- 8.2 EXERCISES, 646
- 8.3 EXERCISES, 647
- 8.4 EXERCISES, 649
- 8.5 EXERCISES, 650
- 8.6 EXERCISES, 652
  - MASTERY SKILLS CHECK: Chapter 8, 655
  - CUMULATIVE SKILLS CHECK: Chapter 8, 659
  - REVIEW SKILLS CHECK: Chapter 9, 662
- 9.1 EXERCISES, 664
- 9.2 EXERCISES, 666
- 9.3 EXERCISES, 668
- 9.4 EXERCISES, 671
- 9.5 EXERCISES, 673
- 9.6 EXERCISE, 674
  - MASTERY SKILLS CHECK: Chapter 9, 677
  - CUMULATIVE SKILLS CHECK: Chapter 9, 682
  - REVIEW SKILLS CHECK: Chapter 10, 684
- 10.2 EXERCISES, 687
- 10.3 EXERCISES, 688
- 10.4 EXERCISE, 689
  - MASTERY SKILLS CHECK: Chapter 10, 694
  - CUMULATIVE SKILLS CHECK: Chapter 10, 694
  - REVIEW SKILLS CHECK: Chapter 11, 698

- 11.1 EXERCISES, 698
- 11.2 EXERCISES, 699
- 11.3 EXERCISES, 701
- 11.4 EXERCISES, 702
- 11.5 EXERCISES, 702
  - MASTERY SKILLS CHECK: Chapter 11, 703
  - REVIEW SKILLS CHECK: Chapter 12, 710
- 12.1 EXERCISES, 712
- 12.2 EXERCISES, 712
- 12.3 EXERCISES, 713
  - MASTERY SKILLS CHECK: Chapter 12, 714
  - CUMULATIVE SKILLS CHECK: Chapter 12, 715
  - REVIEW SKILLS CHECK: Chapter 13, 717
- 13.1 EXERCISES, 717
- 13.2 EXERCISES, 719
- 13.3 EXERCISES, 720
- 13.4 EXERCISES, 723
- 13.5 EXERCISES, 724
- 13.7 EXERCISES, 724
  - MASTERY SKILLS CHECK: Chapter 13, 726
  - CUMULATIVE SKILLS CHECK: Chapter 13, 727
  - REVIEW SKILLS CHECK: Chapter 14, 727
- 14.1 EXERCISES, 728
- 14.3 EXERCISES, 728
- 14.4 EXERCISES, 729
- 14.5 EXERCISES, 732
- 14.6 EXERCISES, 733
- 14.7 EXERCISES, 735
  - MASTERY SKILLS CHECK: Chapter 14, 737

## **Acknowledgments**

I wish to say special thanks to

Bjarne Stroustrup

Steve Clamage

P. J. Plauger

Al Stevens

for sharing their knowledge, advice, and expertise during the preparation of this book. It was much appreciated.

## Introduction

If you already know C and are moving up to C++, this book is for you.

C++ is the C programmer's answer to Object-Oriented Programming (OOP). Built upon the solid foundation of C, C++ adds support for OOP (and many other new features) without sacrificing any of C's power, elegance, or flexibility. C++ has become the universal language of programmers around the world and is the language that will create the next generation of high-performance software. It is the single most important language that a professional programmer must know.

C++ was invented in 1979 by Bjarne Stroustrup at Bell Laboratories in Murray Hill, New Jersey. Initially it was called "C with classes." The name was changed to C++ in 1983. Since then, C++ has undergone three major revisions, the first in 1985 and the second in 1990. The third occurred during the C++ standardization process. Several years ago, work began on a standard for C++. Towards that end, a joint ANSI (American National Standards Institute) and ISO (International Standards Organization) standardization committee was formed. The first draft of the proposed standard was created on January 25, 1994. In that draft, the ANSI/ISO C++ committee (of which I am a member) kept the features first defined by Stroustrup and added some new ones as well. But, in general, this initial draft reflected the state of C++ at the time.

Soon after the completion of the first draft of the standard an event occurred that caused the standard to be greatly expanded: the creation of the Standard Template Library (STL) by Alexander Stepanov. As you will learn, the STL is a set of generic routines that you can use to manipulate data. It is both powerful and elegant. But it is also quite large. Subsequent to the first draft, the committee voted to include the STL in the specification for C++. The addition of the STL expanded the scope of C++ well beyond its original definition. While important, the inclusion of the STL, among other things, slowed the standardization of C++.

It is fair to say that the standardization of C++ took far longer than any one had expected when it began. However, it is now nearly complete. The final draft has been prepared and passed out of

committee. It now awaits only formal approval. In a practical sense, a standard for C++ is now a reality. Compilers already are beginning to support all of the new features.

The material in this book describes Standard C++. This is the version of C++ created by the ANSI/ISO standardization committee, and it is the one that is currently accepted by all major compilers. Therefore, using this book, you can be confident that what you learn today will also apply tomorrow.

### **What Is New in the Third Edition**

This is the third edition of *Teach Yourself C++*. It includes all of the material contained in the first two editions and adds two new chapters and many new topics. The first new chapter covers Run-Time Type ID (RTTI) and the new casting operators. The second covers the Standard Template Library (STL). Both of these topics are major features added to the C++ language since the previous edition was published. New topics include namespaces, the new-style headers, and coverage of the modern-style I/O system. In all, the third edition of *Teach Yourself C++* is substantially larger than its preceding two editions.

### **If You're Using Windows**

If your computer uses Windows and your goal is to write Windows-based programs, then you have chosen the right language to learn. C++ is completely at home with Windows programming. However, none of the programs in this book are Windows programs. Instead, they are console-based programs. The reason for this is easy to understand: Windows programs are, by their nature, large and complex. The overhead required to create even a minimal Windows skeletal program is 50 to 70 lines of code. To write Windows programs that demonstrate the features of C++ would require hundreds of lines of code each. Put simply, Windows is not an appropriate environment in which to learn programming. However, you can still use a Windows-based compiler to compile the programs in this book because the compiler will automatically create a console session in which to execute your program.

Once you have mastered C++, you will be able to apply your knowledge to Windows programming. In fact, Windows programming



using C++ allows the use of class libraries such as MFC, that can greatly simplify the development of a Windows program.

### **How This Book Is Organized**

This book is unique because it teaches you the C++ language by applying mastery learning. It presents one idea at a time, followed by numerous examples and exercises to help you master each topic. This approach ensures that you fully understand each topic before moving on.

The material is presented sequentially. Therefore, be sure to work carefully through the chapters. Each one assumes that you know the material presented in all preceding chapters. At the start of every chapter (except Chapter 1) there is a Review Skills Check that tests your knowledge of the preceding chapter. At the end of each chapter you will find a Mastery Skills Check that checks your knowledge of the material present in the chapter. Finally, each chapter concludes with a Cumulative Skills Check which tests how well you are integrating new material with that presented in earlier chapters. The answers to the book's many exercises are found in Appendix B.

This book assumes that you are already an accomplished C programmer. Put simply, you can't learn to program in C++ until you can program in C. If you can't program in C, take some time to learn it before attempting to use this book. A good way to learn C is to read my book *Teach Yourself C, Third Edition* (Osborne/McGraw-Hill, Berkeley CA, 1997). It uses the same presentation style as this book.

### **Don't Forget: Code on the Web**

Remember, the source code for all of the programs in this book is available free-of-charge on the Web at <http://www.osborne.com>. Downloading this code prevents you from having to type in the examples.



## For Further Study

*Teach Yourself C++, Third Edition* is your gateway into the "Herb Schildt" series of programming books. Here is a partial list of Schildt's other books.

If you want to learn more about C++, then you will find these books especially helpful.

*C++: The Complete Reference*  
*C++ From the Ground Up*  
*Expert C++*

If you want to learn more about C, the foundation of C++, we recommend

*Teach Yourself C*  
*C: The Complete Reference*  
*The Annotated ANSI C Standard*

If you will be developing programs for the Web, you will want to read

*Java: The Complete Reference*

co-authored by Herbert Schildt and Patrick Naughton.

Finally, if you want to program for Windows, we recommend

*Schildt's Windows 95 Programming in C and C++*  
*Schildt's Advanced Windows 95 Programming in C and C++*  
*Windows NT 4 From the Ground Up*  
*MFC Programming From the Ground Up*

When you need solid answers, fast, turn to **Herbert Schildt**, the recognized authority on programming.