

C++ Weekend Crash Course

C++ Weekend Crash Course

Stephen R. Davis



IDG Books Worldwide, Inc.

An International Data Group Company

Foster City, CA ♦ Chicago, IL ♦ Indianapolis, IN ♦ New York, NY

C++ Weekend Crash Course

Published by

IDG Books Worldwide, Inc.

An International Data Group Company

919 E. Hillsdale Blvd., Suite 400

Foster City, CA 94404

www.idgbooks.com (IDG Books Worldwide Web site)

Copyright © 2000 IDG Books Worldwide, Inc. All rights reserved. No part of this book, including interior design, cover design, and icons, may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without the prior written permission of the publisher.

Library of Congress Card Number: 00-101537

ISBN: 0-7645-4689-9

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/RS/QU/QQ/FC

Distributed in the United States by IDG Books Worldwide, Inc.

Distributed by CDG Books Canada Inc. for Canada; by Transworld Publishers Limited in the United Kingdom; by IDG Norge Books for Norway; by IDG Sweden Books for Sweden; by IDG Books Australia Publishing Corporation Pty. Ltd. for Australia and New Zealand; by TransQuest Publishers Pte Ltd. for Singapore, Malaysia, Thailand, Indonesia, and Hong Kong; by Gotop Information Inc. for Taiwan; by ICG Muse, Inc. for Japan; by Intersoft for South Africa; by Eyrolles for France; by International Thomson Publishing for Germany, Austria, and Switzerland; by Distribuidora Cuspide for Argentina; by LR International for Brazil; by Galileo Libros for Chile; by Ediciones ZETA S.C.R. Ltda. for Peru; by WS Computer Publishing Corporation, Inc., for the Philippines; by Contemporanea de Ediciones for

Venezuela; by Express Computer Distributors for the Caribbean and West Indies; by Micronesia Media Distributor, Inc. for Micronesia; by Chips Computadoras S.A. de C.V. for Mexico; by Editorial Norma de Panama S.A. for Panama; by American Bookshops for Finland.

For general information on IDG Books Worldwide's books in the U.S., please call our Consumer Customer Service department at 800-762-2974. For reseller information, including discounts and premium sales, please call our Reseller Customer Service department at 800-434-3422.

For information on where to purchase IDG Books Worldwide's books outside the U.S., please contact our International Sales department at 317-596-5530 or fax 317-572-4002.

For consumer information on foreign language translations, please contact our Customer Service department at 800-434-3422, fax 317-572-4002, or e-mail rights@idgbooks.com.

For information on licensing foreign or domestic rights, please phone +1-650-653-7098.

For sales inquiries and special prices for bulk quantities, please contact our Order Services department at 800-434-3422 or write to the address above.

For information on using IDG Books Worldwide's books in the classroom or for ordering examination copies, please contact our Educational Sales department at 800-434-2086 or fax 317-572-4005.

For press review copies, author interviews, or other publicity information, please contact our Public Relations department at 650-653-7000 or fax 650-653-7500.

For authorization to photocopy items for corporate, personal, or educational use, please contact Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, or fax 978-750-4470.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND AUTHOR HAVE USED THEIR BEST EFFORTS IN PREPARING THIS BOOK. THE PUBLISHER AND AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS BOOK AND SPECIFICALLY DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTIONS CONTAINED IN THIS PARAGRAPH. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES OR WRITTEN SALES MATERIALS. THE ACCURACY AND COMPLETENESS OF THE INFORMATION PROVIDED HEREIN AND THE OPINIONS STATED HEREIN ARE NOT GUARANTEED OR WARRANTED TO PRODUCE ANY PARTICULAR RESULTS, AND THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY INDIVIDUAL. NEITHER THE PUBLISHER NOR AUTHOR SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

Trademarks: All brand names and product names used in this book are trade names, service marks, trademarks, or registered trademarks of their respective owners. IDG Books Worldwide is not associated with any product or vendor mentioned in this book.



® is a registered trademark or trademark under exclusive license to IDG Books Worldwide, Inc. from International Data Group, Inc. in the United States and/or other countries.

ABOUT IDG BOOKS WORLDWIDE

Welcome to the world of IDG Books Worldwide.


IDG Books Worldwide, Inc., is a subsidiary of International Data Group, the world's largest publisher of computer-related information and the leading global provider of information services on information technology. IDG was founded more than 30 years ago by Patrick J. McGovern and now employs more than 9,000 people worldwide. IDG publishes more than 290 computer publications in over 75 countries. More than 90 million people read one or more IDG publications each month.

Launched in 1990, IDG Books Worldwide is today the #1 publisher of best-selling computer books in the United States. We are proud to have received eight awards from the Computer Press Association in recognition of editorial excellence and three from Computer Currents' First Annual Readers' Choice Awards. Our best-selling *...For Dummies®* series has more than 50 million copies in print with translations in 31 languages. IDG Books Worldwide, through a joint venture with IDG's Hi-Tech Beijing, became the first U.S. publisher to publish a computer book in the People's Republic of China. In record time, IDG Books Worldwide has become the first choice for millions of readers around the world who want to learn how to better manage their businesses.

Our mission is simple: Every one of our books is designed to bring extra value and skill-building instructions to the reader. Our books are written by experts who understand and care about our readers. The knowledge base of our editorial staff comes from years of experience in publishing, education, and journalism — experience we use to produce books to carry us into the new millennium. In short, we care about books, so we attract the best people. We devote special attention to details such as audience, interior design, use of icons, and illustrations. And because we use an efficient process of authoring, editing, and desktop publishing our books electronically, we can spend more time ensuring superior content and less time on the technicalities of making books.

You can count on our commitment to deliver high-quality books at competitive prices on topics you want to read about. At IDG Books Worldwide, we continue in the IDG tradition of delivering quality for more than 30 years. You'll find no better book on a subject than one from IDG Books Worldwide.



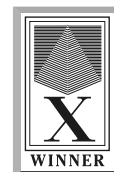

John Kilcullen
Chairman and CEO
IDG Books Worldwide, Inc.



*Eighth Annual
Computer Press
Awards* 1992



*Ninth Annual
Computer Press
Awards* 1993



*Tenth Annual
Computer Press
Awards* 1994



*Eleventh Annual
Computer Press
Awards* 1995

IDG is the world's leading IT media, research and exposition company. Founded in 1964, IDG had 1997 revenues of \$2.05 billion and has more than 9,000 employees worldwide. IDG offers the widest range of media options that reach IT buyers in 75 countries representing 95% of worldwide IT spending. IDG's diverse product and services portfolio spans six key areas including print publishing, online publishing, expositions and conferences, market research, education and training, and global marketing services. More than 90 million people read one or more of IDG's 290 magazines and newspapers, including IDG's leading global brands — Computerworld, PC World, Network World, Macworld and the Channel World family of publications. IDG Books Worldwide is one of the fastest-growing computer book publishers in the world, with more than 700 titles in 36 languages. The *...For Dummies®* series alone has more than 50 million copies in print. IDG offers online users the largest network of technology-specific Web sites around the world through IDG.net (<http://www.idg.net>), which comprises more than 225 targeted Web sites in 55 countries worldwide. International Data Corporation (IDC) is the world's largest provider of information technology data, analysis and consulting, with research centers in over 41 countries and more than 400 research analysts worldwide. IDG World Expo is a leading producer of more than 168 globally branded conferences and expositions in 35 countries including E3 (Electronic Entertainment Expo), Macworld Expo, ComNet, Windows World Expo, ICE (Internet Commerce Expo), Agenda, DEMO, and Spotlight. IDG's training subsidiary, ExecuTrain, is the world's largest computer training company, with more than 230 locations worldwide and 785 training courses. IDG Marketing Services helps industry-leading IT companies build international brand recognition by developing global integrated marketing programs via IDG's print, online and exposition products worldwide. Further information about the company can be found at www.idg.com.

1/26/00

Credits

Acquisitions Editor

Greg Croy

Project Editor

Matthew E. Lusher

Technical Editor

Greg L. Guntle

Copy Editors

S.B. Kleinman

Rich Adin

Media Development Specialist

Jason Lusher

Permissions Editor

Lenora Chin Sell

Media Development Manager

Stephen Noetzel

Project Coordinators

Linda Marousek

Louigene A. Santos

Marcos Vergara

Graphics and Production Specialists

Robert Bihlmayer

Jude Levinson

Michael Lewis

Ramses Ramirez

Victor Pérez-Varela

Dina F Quan

Quality Control Specialist

Laura Taflinger

Proofreading and Indexing

York Production Services

Illustrators

Mary Jo Richards

Brent Savage

Cover Design

Clark Creative Group

About the Author

A 43-year-old father and husband, lives and works in Greenville, Texas as a programmer both at work and at play when he's not riding his bicycle or hauling his son to Tae Kwon Do events.

to my wonderful new nieces, Christa and Sarah

Preface

C++ *Weekend Crash Course* teaches the reader C++ in one admittedly busy weekend: 30 sessions of a half-hour each, for 15 hours stretching from Friday evening to Sunday afternoon. At the end of each part of the book, you'll get a chance to pause, reflect, and review what you've just learned before pushing on through the rest. Good luck!

What is C++?

C++ is the most popular programming language in use today. C++ is used in applications from the micro-programs that drive your microwave oven, your clothes washer and your TV up through the huge, hardcore programs that control nuclear missiles and Mars rockets — heh, you can't blame the Mars rockets on C++.

In the late 1980s C began to show signs of age. For one, C does not support the object-oriented programming style. At the time, the object-oriented wave was taking the world by storm. Employers were throwing money at object-oriented programmers. All you had to do was work the phrase “new paradigm” into the conversation in order to gather a crowd of admirers.

The problem was that every program worth its salt was written in C (there were a few programs written in Pascal like early versions of Windows, but they don't count — if you are familiar with the earliest versions of Windows, you know why). There was no way that companies were going to rewrite all that code just to ride the object-oriented wave.

Object-oriented concepts had to be grafted onto the existing C language. The result was called C++.

C++ is a superset of C. Any well written C program can be rebuilt with a C++ tool to generate a working program. That meant the companies could upgrade their software in pieces. Existing code could remain in C while new code adopted the extra features of C++.

Fortunately for us, C++ is a standardized language. The American National Standards Institute (ANSI) and International Standards Organization (ISO) agree on what C++ is. They issued a detailed description of the C++ language. This standardized language is often known as ANSI or ISO standard C++ or simply Standard C++.

Standard C++ is not controlled by a single company such as Microsoft (or Sun, for that matter). The Standard C++ community is not held hostage to the whims of any one corporate giant. In addition, companies do not stray. Even Microsoft's Visual C++ holds tightly to the C++ standard.

The programs in *C++ Weekend Crash Course* can be built using any Standard C++ implementation.

The Object-Oriented Paradigm

Object-oriented programming is not all hype. Object-oriented programming really is a different approach to programming than its predecessor. Object-oriented programs are easier to write and maintain. Object-oriented modules can be reused with greater ease than those written in older styles.

C++ Weekend Crash Course presents more than just the C++ language. You need to learn the object-oriented paradigm in order to make complete use of the power of C++. *C++ Weekend Crash Course* uses C++ examples to teach you the object-oriented view of the world. Anyone who claims to program in C++ without understanding OO concepts is just using C++ as a “better C”.

Who

C++ Weekend Crash Course is intended for the beginner through the intermediate reader.

This book serves the beginner by not assuming any knowledge of programming or programming concepts. The first few lessons go over real-world, non-techie explanations of what programming is.

This book is also great for the home programmer. The multiple examples demonstrate programming techniques used in modern, high speed programs.

The serious programmer or student needs C++ in his quiver of programming skills. The ability to speak knowledgeably of C++ can make the difference between getting that job and not.

What

C++ Weekend Crash Course is more than just a book: it's a complete development package. A CD-ROM containing the famous GNU C++ environment is included with the book.

You need a word processor, such as Microsoft Word, in order to do word processing. Similarly, you need a C++ development environment in order to build and execute programs in C++.

Many readers will already own a programming environment such as Microsoft's ubiquitous Visual C++. For those who do not own a C++ environment already, *C++ Weekend Crash Course* includes the standard GNU C++.

GNU C++ is not some stripped down, limited time program. The GNU C++ package included with the book is a complete, no-holds-barred development environment. *C++ Weekend Crash Course* provides complete instructions on how to install and use both GNU C++ and Visual C++.

How

C++ Weekend Crash Course follows a one-weekend format. Start with Friday evening; conclude Sunday afternoon.

This "One weekend" format is:

- ideal for the student who wants to catch up with the rest of the class,
- ideal for the one-time programmer who wants to brush up on his skills, and
- ideal for anyone who wants to learn C++ while the kids are off at Grandma's house.

Of course, you can proceed through the book at a more leisurely pace, if you prefer. Each section of 4 to 6 lessons can be read independently.

The reader should be able to complete each of 30 sessions in 30 minutes. Time markers in the lesson margin help keep the reader on pace.

Each session is followed by a set of review questions to allow the reader to judge her comprehension of the material. A set of more involved problems is provided at the end of each part to help drive home knowledge gained during the weekend session.

Overview

C++ Weekend Crash Course presents its sessions in groups of 4 to 6 chapters, organized into 6 parts:

Friday evening — Introduction to programming.

This part introduces programming concepts and progresses you through your first program.

Saturday morning — Basic C++

This part covers beginning topics such as statement syntax, operators and basic function.

Saturday afternoon — Structures and pointers.

Here the reader delves the slightly more complicated topic of pointer variables including their application in linked lists, arrays and objects.

Saturday evening — Introduction to object based programming.

This is the jumping-off point — topics such as C++ structures, which form the basis for object-oriented programming are discussed.

Sunday morning — Object-oriented programming.

Here it is — the mother lode. This part delves into both the syntax and the meaning of object-oriented programming.

Sunday afternoon — Wrap up

This part wraps up some of the more involved topics such as error handling using exceptions and overloading operators.

Each part ends with a discussion of debugging techniques for finding and removing the inevitable errors from your programs. The level of complexity of these techniques is chosen to match the reader's ability glean from that session.

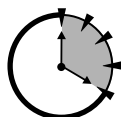
The appendix includes more involved programming problems for each lesson.

Layout and Features

No one should try to simply power through this material without a break. After each session, and at the end of each part, you'll find some questions to check your knowledge and give you a little practice at exercising your new-found skills. Take a break, grab a snack, refill that coffee mug, and plunge into the next one!

Along the way, you'll find some features of the book to help you keep track of how far along you are, and point out interesting bits of info you shouldn't miss. First, as you're going through each session, check for something like this in the margin:

This icon and others like it let you know how much progress you've made through each session as you go. There are also several icons to point out special tidbits of info for you:



**20 Min.
To Go**



This is a flag to clue you in to an important piece of info you should file away in your head for later.



This gives you helpful advice on the best ways to do things, or a neat little technique that can make your programming go easier.



Don't do this! 'Nuff said.



This highlights information you'll find on the CD-ROM that accompanies this book.

We also occasionally highlight text passages that explain key concepts of C++ syntax, like so:

SYNTAX ►

A **function** is a logically separate block of C++ code. The function construct has the form:

```
<return type> name(<arguments to the function>)
{
    // ...
    return <expression>;
}
```

Conventions Used in this Book

Aside from the icons you've just seen, such as *Tip*, there are only three conventions in this book:

- To indicate a menu choice, we use the ⇨ symbol, as in:
Choose File ⇨ Save Project to save your work.
- To indicate programming code within the body text, we use a special font, like this:
Likewise, when writing `main()`, I could concentrate on handling the summation returned by `sumSequence()`, while thinking only of what the function did, and not about how it worked.
- To indicate a programming example that's not in the body text, we use this typeface:

```
float fVariable1 = 10.0;
float fVariable2 = (10 / 3) * 3;
fVariable1 == fVariable2;    // are these two equal?
```

What's left?

Nothing. Open your work book to the first page and start the clock. It's Friday evening: you have two days.

Acknowledgments

Writing a book like C++ Weekend Crash Course is a challenge, especially since it's one of the first titles in a new series. I'm pleased to have had the opportunity to help launch a new way to teach readers the basics of programming.

I'd first like to thank Greg Croy, acquisitions editor, for spearheading this new series and selecting me as an author. I'd also like to thank my agent, Claudette Moore, for her work with Greg and me to get this project moving.

The editorial staff at IDG Books has been very helpful, and their contributions have made this a better book: Matt Lusher, project editor; S.B. Kleinman and Rich Adin, copy editors; and the production staff directly responsible for the look of what you now hold in your hands. Greg Guntle, technical editor, provided a sharp eye for accuracy and detail.

Finally, and most of all, I'd like to thank my family, whose support of my writing makes it all worthwhile.

Contents at a Glance

Preface	ix
Acknowledgments	xv
FRIDAY	2
Part I—Friday Evening	4
Session 1—What Is Programming Anyway?	5
Session 2—Creating Your First C++ Program in Visual C++	11
Session 3—Creating Your First C++ Program in GNU C++	23
Session 4—C++ Instructions	37
Part I—Friday Evening Review	45
SATURDAY	48
Part II—Saturday Morning	50
Session 5—Variable Types	51
Session 6—Mathematical Operations	65
Session 7—Logical Operations	73
Session 8—Flow Control Commands	85
Session 9—Functions	101
Session 10—Debugging I	115
Part II—Saturday Morning Review	127
Part III—Saturday Afternoon	130
Session 11—The Array	131
Session 12—Intro to Classes	149
Session 13—A Few C++ Pointers	159
Session 14—A Few More Pointers	175
Session 15—Pointers to Objects	195
Session 16—Debugging II	213
Part III—Saturday Afternoon Review	229
Part IV—Saturday Evening	232
Session 17—Object Programming	233
Session 18—Active Classes	239
Session 19—Maintaining Class Integrity	257
Session 20—Class Constructors II	273
Part IV—Saturday Evening Review	296

SUNDAY	298
Part V—Sunday Morning	300
Session 21—Inheritance	301
Session 22—Polymorphism	315
Session 23—Abstract Classes and Factoring	329
Session 24—Multiple Inheritance	345
Session 25—Large Programs	361
Session 26—C++ Preprocessor	373
Part V—Sunday Morning Review	384
Part VI—Sunday Afternoon	388
Session 27—Overloading Operators	389
Session 28—The Assignment Operator	409
Session 29—Stream I/O	419
Session 30—Exceptions	441
Part VI—Sunday Afternoon Review	456
Appendix A	461
Appendix B	483
Appendix C	497
Index	501
End-User License Agreement	514
GNU General Public License.....	517
CD-ROM Installation Instructions.....	526

Contents

Preface	ix
Acknowledgments	xv
FRIDAY	2
Part I—Friday Evening	4
Friday Evening	4
Session 1—What Is Programming Anyway?	5
A Human Program	6
The algorithm	6
The processor	7
The program	7
Computer processors	9
Session 2—Creating Your First C++ Program in Visual C++	11
Installing Visual C++	12
Creating Your First Program	12
Building Your Program	14
Executing Your Program	18
Closing Points	20
Program output	20
Visual C++ help	20
Session 3—Creating Your First C++ Program in GNU C++	23
Installing GNU C++	24
Creating Your First Program	26
Entering the C++ code	26
Building Your Program	29
Executing Your Program	33
Closing Points	34
Program output	35
GNU C++ help	35
Session 4—C++ Instructions	37
The Program	37
The C++ Program Explained	39
The basic program framework	39
Comments	40
There's that framework again	40
Statements	41
Declarations	41
Input/output	42

Expressions	42
Assignment	43
Expressions (continued)	43
Part I–Friday Evening Review	45
SATURDAY	48
Part II–Saturday Morning	50
Session 5–Variable Types	51
Decimal Numbers	52
The limitations of int's in C++	52
Solving the truncation problem	55
Limitations of floating point	56
Other Variable Types	57
Types of constants	59
Special characters	59
Mixed Mode Expressions	61
Session 6–Mathematical Operations	65
Arithmetic Operators	66
Expressions	67
Operator Precedence	68
Unary Operators	69
Assignment Operators	71
Session 7–Logical Operations	73
Simple Logical Operators	74
Short circuits and C++	76
Logical variable types	76
Binary Numbers	77
Bitwise Logical Operations	78
The single-bit operators	78
The bitwise operators	80
A simple test	81
Why?	82
Session 8–Flow Control Commands	85
The Branch Command	86
Looping Commands	87
The while loop	88
The for loop	91
Special loop controls	93
Nested Control Commands	96
Can We switch to a Different Subject?	97

Session 9—Functions	101
Sample Function Code	102
Sample code	102
Function	105
Why use functions?	105
Simple functions	106
Functions with arguments	107
Multiple functions with the same nickname	110
Function Prototypes	112
Variable Storage Types	113
Session 10—Debugging I	115
Types of Errors	115
The Output-Statement Technique	116
Catching Bug #1	118
Visual C++	118
GNU C++	120
Catching Bug #2	122
Part II—Saturday Morning Review	127
Part III—Saturday Afternoon	130
Session 11—The Array	131
What Is an Array?	132
An array in practice	134
Initializing an array	137
Why use arrays?	138
Arrays of arrays	138
Arrays of Characters	139
Manipulating Strings	142
Our own concatenate function	142
C++ string-handling functions	145
Wide characters	146
Obsolescent Output Functions	146
Session 12—Intro to Classes	149
Grouping Data	149
An example	150
The problem	153
The Class	153
The format of a class	153
Example program	155
Advantages	157

Session 13—A Few C++ Pointers	159
What's Your Address?	160
Introduction to Pointer Variables	162
Types of Pointers	165
Passing Pointers to Functions	167
Pass by value	167
Passing pointer values	168
Passing by reference	169
Heap Memory	169
Scope	170
The scope problem	171
The heap solution	172
Session 14—A Few More Pointers	175
Pointers and Arrays	176
Operations on pointers	176
Character arrays	180
Operations on different pointer types	185
Differences between pointers and arrays	185
Arguments to the Program	187
Arrays of pointers	187
Arrays of character strings	188
The arguments to main()	190
Session 15—Pointers to Objects	195
Pointers to Objects	195
Passing objects	196
References	198
Return to the heap	199
The Array Data Structure	199
Linked Lists	200
Adding to the head of a linked list	202
Other operations on a linked list	203
Properties of linked lists	205
A Linked NameData Program	206
Other Containers	210
Session 16—Debugging II	213
Which Debugger?	214
The Test Program	215
Single-Stepping Through a Program	217
Single-Stepping into a Function	218
Using Breakpoints	220
Viewing and Modifying Variables	221
Using the Visual C++ Debugger	226
Part III—Saturday Afternoon Review	229

Part IV—Saturday Evening	232
Session 17—Object Programming	233
Abstraction and Microwave Ovens	233
Functional nachos	234
Object-oriented nachos	235
Classification and Microwave Ovens	235
Why Build Objects This Way?	236
Self-Contained Classes	237
Session 18—Active Classes	239
Class Review	240
Limitations of Struct	241
A functional fix	241
Defining an Active Class	243
Naming member functions	244
Defining a Member Function in the Class	245
Writing Member Functions Outside of the Class	247
Include files	248
Calling a Member Function	249
Calling a member function with a pointer	250
Accessing other members from a member function	251
Overloading Member Functions	253
Session 19—Maintaining Class Integrity	257
Creating and Destroying Objects	257
The constructor	259
The destructor	264
Access Control	266
The protected keyword	267
Static data members	270
Session 20—Class Constructors II	273
Constructors with Arguments	273
Constructing Class Members	277
Order of Construction	283
Local objects are constructed in order	284
Static objects are constructed only once	284
All global objects are constructed before main()	285
Global objects are constructed in no particular order	285
Members are constructed in the order in which they are declared	286
Destructors are invoked in the reverse order of the constructors	287
The Copy Constructor	287
Shallow copies versus deep copies	290
A “fall back” copy constructor	291

Exercises	293
Problems	293
Hint	294
My solution	294
Part IV–Saturday Evening Review.....	296
SUNDAY	298
Part V–Sunday Morning	300
Session 21–Inheritance	301
Advantages of Inheritance	302
Class Factoring	303
Implementing Inheritance in C++	303
Constructing a Subclass	308
The HAS_A Relationship	311
Session 22–Polymorphism	315
Overriding Member Functions	316
Enter Polymorphism	318
Polymorphism and Object-Oriented Programming	320
How Does Polymorphism Work?	322
When Is a Virtual Function Not?	324
Virtual Considerations	326
Session 23–Abstract Classes and Factoring	329
Factoring	329
Abstract Classes	334
Declaring an abstract class	335
Making a “concrete” class out of an abstract class	337
When is a subclass abstract?	339
Passing an abstract object to a function	341
Why are pure virtual functions needed?	342
Session 24–Multiple Inheritance	345
How Does Multiple Inheritance Work?	346
Inheritance Ambiguities	348
Virtual Inheritance	349
Constructing the Objects of Multiple Inheritance	356
A Contrary Opinion	357
Session 25–Large Programs	361
Why Divide Programs?	361
Separating Class Definition from Application Program	362
Dividing the program	362
The #include directive	364
Dividing application code	365
Project file	367
Reexamining the standard program template	370
Handling outline member functions	371

Session 26—C++ Preprocessor	373
The C++ Preprocessor	373
The #include Directive	374
The #define Directive	375
Defining macros	376
Common errors using macros	376
Compile Controls	378
The #if directive	378
The #ifdef directive	379
Part V—Sunday Morning Review.....	384
Part VI—Sunday Afternoon	388
Session 27—Overloading Operators	389
Why Do I Need to Overload Operators?	390
What Is the Relationship Between	
Operators and Functions?	391
How Does Operator Overloading Work?	391
Special considerations	395
A More Detailed Look	395
Operators as Member Functions	398
Yet Another Overloading Irritation	401
When Should an Operator be a Member?	402
Cast Operator	402
Session 28—The Assignment Operator	409
Why Is Overloading the Assignment Operator Critical?	409
Comparison with copy constructor	410
How Do I Overload the Assignment Operator?	411
Two more details about the assignment operator	415
An Escape Hatch	416
Session 29—Stream I/O	419
How Does Stream I/O Work?	420
The fstream Subclasses	421
The strstream Subclasses	425
Comparison of string-handling techniques	427
Manipulators	431
Custom Inserters	433
Smart Inserters	435
But Why the Shift Operators?	438
Session 30—Exceptions	441
Conventional Error Handling	441
Why Do I Need a New Error Mechanism?	442

How Do Exceptions Work?	444
Why is the exception mechanism an improvement?	446
Trying to Catch the Details Thrown at You	446
What kinds of things can I throw?	449
Chaining Catch Phrases	453
Part VI—Sunday Afternoon Review.....	456
 Appendix A—Answers to Part Reviews	 461
Appendix B—Supplemental Problems	483
Appendix C—What’s On the CD-ROM	497
Index	501
End-User License Agreement	514
GNU General Public License	517
CD-ROM Installation Instructions	526

