



# Object-Oriented Programming Using C++

Third Edition

Joyce Farrell

---

---

# Object-Oriented Programming Using C++ Third Edition

Joyce Farrell

**THOMSON**  
★  
**COURSE TECHNOLOGY**™

Australia • Canada • Mexico • Singapore • Spain • United Kingdom • United States

---



*Object-Oriented Programming Using C++, Third Edition* is published by Thomson Course Technology.

**Executive Editor**  
Bob Woodbury

**Managing Editor**  
William Pitkin III

**Senior Product Manager**  
Tricia Coia

**Production Editor**  
Danielle Chouhan

**Developmental Editor**  
Lisa Ruffolo

**Associate Product Manager**  
Sarah Santoro

**Product Marketing Manager**  
Brian Berkeley

**Editorial Assistant**  
Allison Murphy

**Senior Manufacturing Coordinator**  
Justin Palmeiro

**Cover Designer**  
Steve Deschene

**Compositor**  
GEX Publishing Services

**Copyeditor**  
Gary Michael Stahl

**Proofreader**  
John Bosco

**Indexer**  
Sharon Hilgenberg

COPYRIGHT 2007 Thomson Course Technology, a division of Thomson Learning. Thomson Learning is a trademark used herein under license.

Printed in the United States of America.

1 2 3 4 5 CS 10 09 08 07 06

#### **Disclaimer**

Thomson Course Technology reserves the right to revise this publication and make changes from time to time in its content without notice.

For more information, contact Thomson Course Technology, 25 Thomson Place, Boston, MA 02210; or find us on the World Wide Web at [www.course.com](http://www.course.com).

ALL RIGHTS RESERVED. No part of this work may be reproduced, transcribed, or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, Web distribution, or information storage and retrieval systems—without prior written permission of the publisher.

The Web addresses in this book are subject to change from time to time as necessary without notice.

For permission to use material from this text or product, submit a request online at [www.thomsonrights.com](http://www.thomsonrights.com).

Any additional questions about permissions can be submitted by e-mail to [thomsonrights@thomson.com](mailto:thomsonrights@thomson.com).

ISBN 1-4188-3626-5

---

# Preface

---

**O**bject-Oriented Programming Using C++, Third Edition is designed for many levels of programming students and a variety of programming teaching styles. Readers who are new to programming will find the basics of programming logic and the C++ programming language covered thoroughly and clearly. Clear, thorough explanations, multiple programming examples, and step-by-step programming lessons provide beginning readers with a solid C++ background. Users who know some C++ syntax, but are new to object-oriented programming, will find objects explored thoroughly from the first chapters. Objects are introduced early, so those who want to learn objects at the start of their programming experience can do so. Users who want to postpone objects can simply omit the later sections of each chapter and cover the basic programming structures with simple data types before returning to the more complex objects later on.

## Organization and Coverage

*Object-Oriented Programming Using C++* contains 14 chapters and five appendices that present clear text explanation, directed hands-on instruction, and a wealth of exercises. In these chapters, readers learn about programming logic in general, C++ syntax in particular, and gain an appreciation for and understanding of the object-oriented approach. When readers complete the book, they will have an understanding of object-oriented concepts as they apply to programming, and the ability to use these concepts to develop C++ programs.

Chapter 1 provides an overview of programming in general and C++ in particular. You work with variables, comments, input and output, and data structures. This book distinguishes itself from other C++ books by introducing structure objects in Chapter 1 so that students start thinking in an object-oriented manner from the beginning of the course.

Chapter 2 focuses on evaluating C++ expressions. Chapters 3, 4, and 5 discuss decisions, loops, and arrays, strings, and pointers—all fundamental building blocks of C++ programs. Chapter 6 provides a solid foundation in writing and using functions including passing parameters by value and by reference and returning values from functions.

Once students understand C++ basics they are ready for Chapters 7 and 8, which delve more completely into the object-oriented aspects of C++, featuring classes, objects, and design issues. Friend functions and operator overloading are covered in Chapter 9, and inheritance, another important OO feature, is explained in Chapter 10.

---

Advanced C++ features such as templates, exception handling, and advanced input and output techniques, including writing objects to files, are covered in Chapters 11, 12, and 13. Chapter 14 presents some interesting adjuncts to C++ that make it such a powerful language, including creating enumerations, working with bits, and understanding recursion.

Five appendices offer further explanation to topics mentioned in the chapters. Appendices A and B describe how to get started with various C++ compilers. Appendix C contains a handy table of precedence and associativity. Appendix D contains information on formatting output, and Appendix E is a lesson in generating random numbers—an important skill in creating scientific simulations as well as in creating games.

## Approach

*Object-Oriented Programming Using C++* teaches object-oriented concepts using C++ as a tool to demonstrate these concepts. This book teaches programming concepts using a task-driven rather than a command-driven approach. Structures are introduced in Chapter 1 so that students start thinking about objects right from the start. However, discussion of objects is reserved for the last sections of the first six chapters, so that instructors who prefer to start with a procedural approach can omit these sections at first, then go back to cover them after the first six chapters have been completed.

## Features

*Object-Oriented Programming Using C++* is an exceptional textbook because it also includes the following features:

- **Objectives.** A brief list of objectives appears at the beginning of each chapter so the student has an overview of the main topics to be covered.
- **Tips.** These provide additional information about a procedure or topic, such as an alternative method of performing a procedure.
- **Figures.** Each chapter averages over 30 figures that contain code, working programs, or screen shots of the programs' execution.
- **You Do It.** After students study each chapter's concepts, they are invited to create small applications that illustrate the concepts. Each application is explained step-by-step as the students add appropriate code to interesting applications.
- **Summaries.** A summary that recaps the programming concepts and commands covered follows each chapter.
- **Key Terms.** Each chapter contains a list of all the key terms defined in the chapter, along with explanation and presented in the order covered in the chapter. The list of Key Terms serves as an additional chapter summary.

- **Review Questions.** Each chapter includes 20 multiple choice review questions that test students' understanding of what they learned in the chapter.
- **Exercises.** Each chapter contains interesting exercises that provide students with the opportunity to apply the concepts they have mastered by writing C++ programs.
- **Debugging Exercises.** Each chapter ends with debugging exercises—programs with a few syntax or logical errors. The student can find the errors and fix them, developing crucial skills of reading others' programs, analyzing error messages and probable cause of errors, and solving problems.
- **Running Cases.** The book contains two running cases in which the student develops large classes, adding appropriate features as each new concept is introduced. By the end of the book the student has created two substantial working classes.

## Teaching Tools

The following teaching tools are available when this book is used in a classroom setting. All of the teaching tools available with this book are provided to the instructor on a single CD-ROM.

- **Instructor's Manual.** Additional instructional material to assist in class preparation, including suggestions for lecture topics, including a 14 and 16 week syllabus outline.
  - **Solution Files.** Solutions to all end-of-chapter materials. (Due to the nature of programming, students' solutions may differ from these solutions and still be correct.)
  - **Data Files.** Data files, containing all of the data that students will use for the exercises in this book, are provided on the Thomson Course Technology Web site and on the Instructor's Resource CD-ROM. Additionally, all complete programs that are presented as figures in the book are included so that students can execute the files and experiment with them without a lot of tedious typing. See the "Read This Before You Begin" section for more information on Data Files.
  - **ExamView Test Bank.** This textbook is accompanied by ExamView, a powerful testing software package that allows instructors to create and administer printed, computer (LAN-based), and Internet exams. ExamView includes hundreds of questions that correspond to the topics covered in this text, enabling students to generate detailed study guides that include page references for further review. The computer-based and Internet testing components allow students to take exams at their computers, and also save time for the instructor by grading each exam automatically.
  - **PowerPoint Presentations.** This book comes with Microsoft PowerPoint slides for each chapter. These are included as a teaching aid for classroom presentation, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics they introduce to the class.
-

## ACKNOWLEDGMENTS

I would like to thank all of the people who have made this book a reality, especially Lisa Ruffolo, my Developmental Editor who has kept up her good humor through the last two editions of this book as well as several others. She makes me look good. Thanks also to Tricia Coia, Senior Product Manager, whose help and friendship have been immeasurable not just on this book, but for the decade I have been associated with Thomson Course Technology. Thanks to Danielle Chouhan, Production Editor, Mary Franz, Managing Editor, and Will Pitkin, Managing Editor, as well as the testers in Quality Assurance who play a big part in making sure we produce a high-quality programming text.

Thank you to the reviewers who provided helpful and insightful comments during the development of this book, including Steve Chadwick, Embry-Riddle Aeronautical University; Robert Dollinger, University of Wisconsin Stevens Point; Stephen P. Leach, Florida State University; Paul Turnage, Schoolcraft College; Catherine Wyman, DeVry University.

Thank you to my husband Geoff who runs all the errands so I can compile just one more program.

Finally, this book is dedicated to Jann Patton, my cousin and friend.

---

---

# Read This Before You Begin

---

---

## TO THE USER

### Data Files

To complete some of the steps and exercises in this book, you will need data files that have been created for this book. Your instructor will provide the data files to you. You also can obtain the files electronically from the Thomson Course Technology Web site by connecting to [www.course.com](http://www.course.com) and then searching for this book by title, author, or ISBN.

Each chapter in this book has its own set of data files, which is stored in a separate folder. For example, the files for Chapter 3 are stored in the Chapter 3 folder. You can use a computer in your school lab or your own computer to complete the labs and exercises in this book.

### Using Your Own Computer

To use your own computer to complete the steps and exercises in this book, you will need the following:

- A personal computer. This book was written and quality assurance tested using Microsoft Windows 2000 and XP and Linux.
  - A C++ compiler. Almost all examples in this book will work with any C++ compiler. This book was written using Microsoft Visual Studio 2005 Express Edition (beta) and quality assurance tested using Linux and Microsoft Visual C++ .NET 2003 Standard Edition, as well as Microsoft Visual C++ 2005 Express Edition (final). If your book came with a copy of Microsoft Visual Basic 2005 Express Edition, then you may install that on your computer and use it to complete the material. If you purchased a copy of the text, then you also received Microsoft Visual C++ 2005 Standard Edition on CD-ROM. Appendix A contains instructions on getting started with Microsoft Visual Studio. Appendix B contains instructions on getting started with some other compilers and suggests minor modifications you might have to make to your programs to get them to work correctly using different compilers.
  - Data files. You will not be able to complete the steps and exercises in this book using your own computer unless you have the data files. You can get the data files from your instructor, or you can obtain the data files electronically from the Thomson Course Technology Web site by connecting to [www.course.com](http://www.course.com) and then searching for this book title.
-

## Visit Our World Wide Web Site

Additional materials might be available for your course on the Web. Visit the Thomson Course Technology Web site—[www.course.com](http://www.course.com)—and periodically search this site for more details.

---

## TO THE INSTRUCTOR

To complete the labs and exercises in this book, your students must use a set of data files. These files are included on the Teaching Tools CD-ROM. They may also be obtained electronically through the Thomson Course Technology Web site at [www.course.com](http://www.course.com). Follow the instructions in the Help file to copy the data files to your server or standalone computer. You can view the Help file using a text editor such as WordPad or Notepad. Once the files are copied, you should instruct your users how to copy the files to their own computers or workstations.

## Thomson Course Technology Data Files

You are granted a license to copy the data files to any computer or computer network used by individuals who have purchased this book.

---

BRIEF

# Contents

PREFACE	xv
CHAPTER ONE An Overview of Object-Oriented Programming and C++	1
CHAPTER TWO Evaluating C++ Expressions	51
CHAPTER THREE Making Decisions	81
CHAPTER FOUR Performing Loops	119
CHAPTER FIVE Understanding Arrays, Strings, and Pointers	159
CHAPTER SIX Using C++ Functions	209
CHAPTER SEVEN Using Classes	265
CHAPTER EIGHT Class Features and Design Issues	313
CHAPTER NINE Understanding Friends and Overloading Operators	363
CHAPTER TEN Understanding Inheritance	423
CHAPTER ELEVEN Using Templates	475
CHAPTER TWELVE Handling Exceptions	519
CHAPTER THIRTEEN Advanced Input and Output	565
CHAPTER FOURTEEN Advanced Topics	613

---

APPENDIX A	
Getting Started with Microsoft Visual Studio 2005	665
APPENDIX B	
Getting Started with Other C++ Compilers	671
APPENDIX C	
Operator Precedence and Associativity	675
APPENDIX D	
Formatting Output	679
APPENDIX E	
Generating Random Numbers	687
GLOSSARY	693
INDEX	707

---

TABLE OF

# Contents

---

## PREFACE

xv

## CHAPTER ONE

### An Overview of Object-Oriented Programming and C++

1

The Task of Programming	2
Programming Universals	3
Procedural Programming	4
Early Procedural Programs	5
Modularity and Abstraction	8
Encapsulation	10
Object-Oriented Programming	11
Objects and Classes	12
Inheritance	12
Polymorphism	13
Getting Started in the C++ Programming Environment	13
Creating a <code>main()</code> Function	14
Working with Variables and the <code>const</code> Qualifier	16
The <code>int</code> Data Type	18
The <code>char</code> Data Type	19
The <code>bool</code> Data Type	19
Floating-Point Data Types	20
Declaring Variables	20
The <code>const</code> Qualifier	23
Creating Comments	23
ANSI/ISO Standard C++	24
Using Libraries, Preprocessor Directives, and <code>namespace</code>	24
Producing C++ Output	26
Providing C++ Input	28
A First Look at Data Structures and Classes	30
You Do It	33
Creating a Program That Displays Variable Values	33
Introducing Errors into a Program	35
Modifying a Program to Accept Input Values	35
Creating a Simple Structure	36
Chapter Summary	38
Key Terms	39
Review Questions	43
Exercises	46
Case Project 1	49
Case Project 2	49

## CHAPTER TWO

### Evaluating C++ Expressions

51

Using C++ Binary Arithmetic Operators	52
Using Modulus	57
Precedence and Associativity of Arithmetic Operations	58

---

Shortcut Arithmetic Operators	59
Compound Assignment Operators	59
Increment and Decrement Operators	60
Other Unary Operators	62
Evaluating Boolean Expressions	64
Performing Operations on struct Fields	65
You Do It	67
Using Arithmetic Operators	67
Using Prefix and Postfix Increment and Decrement Operators	68
Using Operators with struct Fields	69
Chapter Summary	71
Key Terms	71
Review Questions	73
Exercises	76
Case Project 1	78
Case Project 2	79

## CHAPTER THREE

### Making Decisions

**81**

Using the <code>if</code> Statement	82
The Single-Alternative <code>if</code>	82
The Dual-Alternative <code>if</code>	86
Using a Nested <code>if</code>	87
Avoiding Common Pitfalls with <code>if</code> Statements	88
Pitfall: Forgetting that C++ is case sensitive	88
Pitfalls: Assuming that indentation has a logical purpose, adding an unwanted semicolon, and forgetting curly braces	88
Pitfall: Using <code>=</code> instead of <code>==</code>	90
Pitfall: Making unnecessary comparisons	91
Pitfall: Creating unreachable code	93
Using the <code>switch</code> Statement	94
Using the Conditional Operator	95
Using the Logical AND and the Logical OR	96
Using the Logical AND	96
Using the Logical OR	99
Pitfall: Using OR when you mean AND	101
Combining AND and OR Selections	101
Making Decisions with Structure Fields	102
You Do It	104
Using a Single-Alternative <code>if</code>	104
Using a Dual-Alternative <code>if</code>	105
Using a Compound Condition and Nested <code>ifs</code>	105
Chapter Summary	108
Key Terms	108
Review Questions	109
Exercises	114
Case Project 1	116
Case Project 2	117

---

<b>CHAPTER FOUR</b>	
<b>Performing Loops</b>	<b>119</b>
The <code>while</code> Loop	120
Writing Typical Loops	123
A Typical Loop: Input Verification	123
A Typical Loop: Reading Input Records	125
Avoiding Common Pitfalls with Loops	127
Pitfall: Adding an unwanted semicolon	127
Pitfalls: Forgetting curly braces or failing to alter a loop control variable	128
Pitfall: Failing to initialize a loop control variable	130
Accumulating Totals	131
The <code>for</code> Loop	133
Pretest vs. Posttest Loops	136
Nested Loops	140
Using Loops with Structure Fields	143
You Do It	143
Using a Loop to Validate User Data Entry	143
Using a Structure in an Application Containing Several Loops	144
Chapter Summary	147
Key Terms	148
Review Questions	149
Exercises	154
Case Project 1	156
Case Project 2	157
<b>CHAPTER FIVE</b>	
<b>Understanding Arrays, Strings, and Pointers</b>	<b>159</b>
Understanding Memory Addresses	160
Understanding Arrays	161
Storing Values in an Array	164
Accessing and Using Array Values	165
Avoiding Common Array Errors	168
Pitfall: Forgetting that arrays are zero-based	168
Pitfall: Accessing locations beyond the array	169
Using Part of an Array	170
Using Parallel Arrays	172
Creating Arrays of Structure Objects	176
Using Strings	177
Strings Created as Arrays of Characters	178
Special String-Handling Problems When Using Character Arrays	179
An Introduction to the <code>string</code> Class	186
Using Pointers	188
Using a Pointer Instead of an Array Name	190
You Do It	193
Using an Array	193
Understanding Memory Addresses	195
Chapter Summary	197
Key Terms	198
Review Questions	199
Exercises	203
Case Project 1	206
Case Project 2	207

---

**CHAPTER SIX****Using C++ Functions****209**

Writing Simple Functions	210
Placing Functions within Files	212
Placing a Function as Part of the Same File, Before <code>main()</code>	212
Placing a Function as Part of the Same File, After <code>main()</code>	213
Placing a Function in Its Own File	215
Understanding Procedural Abstraction	218
Understanding Scope	220
Distinguishing between Local and Global Variables	220
Using the Scope Resolution Operator	223
Returning Values from Functions	225
Passing Values to Functions	228
Avoiding Common Errors When Using Functions	231
Pitfall: Neglecting to make sure the function declaration, header, and call agree	231
Pitfall: Indicating a parameter type in a function call	232
Pitfall: Indicating a return type in a function call	232
Pitfall: Ignoring the order of parameters	232
Pitfall: Assuming that an unused return value has an effect	233
Using Objects as Arguments to Functions and as Return Types of Functions	233
Passing Addresses to Functions	235
Using Reference Variables with Functions	238
Declaring Reference Variables	238
Passing Variable Addresses to Reference Variables	239
Passing Arrays to Functions	242
Using Inline Functions	244
Using Default Parameters	245
Overloading Functions	247
Avoiding Ambiguity	249
You Do It	250
Writing Functions That Return Values	250
Writing a Function That Requires a Parameter	252
Chapter Summary	253
Key Terms	254
Review Questions	256
Exercises	259
Case Project 1	262
Case Project 2	263

**CHAPTER SEVEN****Using Classes****265**

Creating Classes	266
Encapsulating Class Components	268
Designing Classes	269
Implementing Class Functions	271
Using Public Functions to Alter Private Data	272
Using Private Functions and Public Data	277
Considering Scope When Defining Member Functions	280
Using Static Class Members	282
Defining Static Data Members	283
Using Static Functions	286

---

Understanding the <code>this</code> Pointer	288
Using the <code>this</code> Pointer Explicitly	291
Using the Pointer-to-Member Operator	291
Understanding Polymorphism	293
You Do It	295
Creating and Using a Class	295
Using a <code>static</code> Field	297
Understanding How <code>static</code> and Non- <code>static</code> Fields are Stored	300
Chapter Summary	301
Key Terms	302
Review Questions	303
Exercises	307
Case Project 1	310
Case Project 2	311

## CHAPTER EIGHT

### **Class Features and Design Issues** **313**

Classifying the Roles of Member Functions	314
Understanding Constructors	314
Writing Constructors without Arguments	316
Writing Constructors with Arguments	320
Pitfall: Using Parentheses When Instantiating an Object with a Default Constructor	323
Overloading Constructors	323
Using Destructors	326
Understanding Composition	330
Using Composition When Member Classes Contain Non-Default Constructors	333
Using <code>#ifndef</code> , <code>#define</code> , and <code>#endif</code>	336
Improving Classes and Their Functions	338
Selecting Member Data and Function Names	339
Reducing Coupling Between Functions	341
Increasing Cohesion Within a Function	341
You Do It	343
Creating a Class with a Constructor	343
Using Constructor Parameters	345
Understanding Composition	346
Chapter Summary	350
Key Terms	352
Review Questions	353
Exercises	356
Case Project 1	360
Case Project 2	361

## CHAPTER NINE

### **Understanding Friends and Overloading Operators** **363**

What are <code>friends</code> ?	364
How to Declare a Function as a <code>friend</code>	365
Understanding the Benefits of Overloading and Polymorphism	369
Using a friend Function to Access Data from Two Classes	371
Using a Forward Reference	373
Overloading Operators—The Rules	375

---

Overloading an Arithmetic Operator	379
Paying Attention to the Order of the Operands	382
Overloading an Operator to Work with a Class Object and a Primitive Type	383
Using Multiple Operations in a Statement	385
Overloading Output	388
Overloading Input	391
Overloading the Prefix ++ and --	394
Using Postfix Increment and Decrement Operators	396
Overloading the == Operator	397
Overloading the = Operator	398
Overloading [ ] and ( )	404
You Do It	407
Overloading an Arithmetic Operator	407
Overloading an Output Operator	409
Chapter Summary	410
Key Terms	411
Review Questions	413
Exercises	416
Case Project 1	421
Case Project 2	422

## CHAPTER TEN

### Understanding Inheritance

**423**

Understanding Inheritance	424
Understanding the Advantages Provided by Inheritance	425
Creating a Derived Class	426
Understanding Inheritance Restrictions	430
Choosing the Class Access Specifier	433
Overriding Inherited Access	435
Overriding and Overloading Parent Class Functions	439
Providing for Base Class Construction	446
Using Multiple Inheritance	450
Disadvantages of Using Multiple Inheritance	453
Using Virtual Base Classes	454
You Do It	455
Creating a Base Class	455
Creating a Child Class	458
Creating Another Child Class	460
Using Multiple Inheritance	461
Chapter Summary	463
Key Terms	465
Review Questions	465
Exercises	469
Case Project 1	471
Case Project 2	473

## CHAPTER ELEVEN

### Using Templates

**475**

Understanding the Usefulness of Function Templates	476
Creating Function Templates	477
Using Multiple Arguments to Function Templates	479
Overloading Function Templates	482
Using More than One Type in a Function Template	484