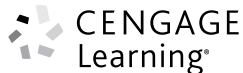


EIGHTH EDITION

# C++ PROGRAMS TO ACCOMPANY PROGRAMMING LOGIC AND DESIGN

BY JO ANN SMITH



Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

**C++ Programs to Accompany  
Programming Logic and Design  
Eighth Edition  
Jo Ann Smith**

Senior Product Manager: Jim Gish

Senior Content Developer: Alyssa Pratt

Content Project Manager:  
Jennifer Feltri-George

Product Assistant: Gillian Daniels

Art Director: Cheryl Pearl, GEX

Proofreader: Lisa Weidenfeld

Copyeditor: Mark Goodin

Indexer: Sharon Hilgenberg

Compositor: Integra Software Services

Cover Designer: GEX Publishing Services

Image Credit: © Kasia/Shutterstock.com

© 2015 Cengage Learning

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at  
**Cengage Learning Customer & Sales Support, [cengage.com/support](http://cengage.com/support)**

For permission to use material from this text or product,  
submit all requests online at **[www.cengage.com/permissions](http://www.cengage.com/permissions)**

Further permissions questions can be e-mailed to  
**[permissionrequest@cengage.com](mailto:permissionrequest@cengage.com)**

Library of Congress Control Number: 2013953075

ISBN-13: 978-1-285-86741-0

**Cengage Learning**  
20 Channel Center Street  
Boston, MA 02210  
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:  
**[www.international.cengage.com/region](http://www.international.cengage.com/region)**

Cengage Learning products are represented in Canada by  
Nelson Education, Ltd.

Visit our corporate Web site at **[cengage.com](http://cengage.com)**.

The programs in this book are for instructional purposes only.

They have been tested with care, but are not guaranteed for any particular intent beyond educational purposes. The author and the publisher do not offer any warranties or representations, nor do they accept any liabilities with respect to the programs.

# Brief Contents

	Preface . . . . .	.x
	Read This Before You Begin . . . . .	.xiii
<b>CHAPTER 1</b>	An Introduction to C++ and the C++ Programming Environment . . . . .	<b>1</b>
<b>CHAPTER 2</b>	Variables, Constants, Operators, and Writing Programs Using Sequential Statements . . . . .	<b>11</b>
<b>CHAPTER 3</b>	Writing Structured C++ Programs . . . . .	<b>29</b>
<b>CHAPTER 4</b>	Writing Programs that Make Decisions . . . . .	<b>43</b>
<b>CHAPTER 5</b>	Writing Programs Using Loops . . . . .	<b>69</b>
<b>CHAPTER 6</b>	Using Arrays in C++ Programs . . . . .	<b>93</b>
<b>CHAPTER 7</b>	File Handling and Applications . . . . .	<b>109</b>
<b>CHAPTER 8</b>	Advanced Array Techniques . . . . .	<b>125</b>
<b>CHAPTER 9</b>	Advanced Modularization Techniques . . . . .	<b>141</b>
<b>CHAPTER 10</b>	Object-Oriented C++ . . . . .	<b>175</b>
	Index . . . . .	<b>195</b>



# Contents

	Preface . . . . .	.x
	Read This Before You Begin . . . . .	.xiii
<b>CHAPTER 1</b>	<b>An Introduction to C++ and the C++ Programming Environment . . . . .</b>	<b>1</b>
	The C++ Programming Language . . . . .	2
	An Introduction to Object-Oriented Terminology . . . . .	2
	The Structure of a C++ Program . . . . .	4
	The C++ Development Cycle . . . . .	6
	Writing C++ Source Code . . . . .	6
	Compiling a C++ Program . . . . .	7
	Executing a C++ Program . . . . .	8
	Exercise 1-1: Understanding the C++ Compiler . . . . .	9
	Lab 1-1: Compiling and Executing a C++ Program . . . . .	9
<b>CHAPTER 2</b>	<b>Variables, Constants, Operators, and Writing Programs Using Sequential Statements . . . . .</b>	<b>11</b>
	Variables . . . . .	12
	Variable Names . . . . .	12
	C++ Data Types . . . . .	13
	Exercise 2-1: Using C++ Variables, Data Types, and Keywords . . . . .	14
	Declaring and Initializing Variables . . . . .	14
	Exercise 2-2: Declaring and Initializing C++ Variables . . . . .	16
	Lab 2-1: Declaring and Initializing C++ Variables . . . . .	16
	Constants . . . . .	17
	Unnamed Constants . . . . .	17
	Named Constants . . . . .	17
	Exercise 2-3: Declaring and Initializing C++ Constants . . . . .	17
	Lab 2-2: Declaring and Initializing C++ Constants . . . . .	18
	Arithmetic and Assignment Operators . . . . .	18
	Arithmetic Operators . . . . .	18
	Assignment Operators and the Assignment Statement . . . . .	19
	Precedence and Associativity . . . . .	20
	Exercise 2-4: Understanding Operator Precedence and Associativity . . . . .	22

	Lab 2-3: Understanding Operator Precedence and Associativity . . . . .	22
	Sequential Statements, Comments, and Interactive Input Statements . . . . .	23
	Exercise 2-5: Understanding Sequential Statements . . . . .	26
	Lab 2-4: Understanding Sequential Statements . . . . .	27
<b>CHAPTER 3</b>	<b>Writing Structured C++ Programs . . . . .</b>	<b>29</b>
	Using Flowcharts and Pseudocode to Write a C++ Program . . . . .	30
	Lab 3-1: Using Flowcharts and Pseudocode to Write a C++ Program . . . . .	34
	Writing a Modular Program in C++ . . . . .	35
	Lab 3-2: Writing a Modular Program in C++ . . . . .	40
<b>CHAPTER 4</b>	<b>Writing Programs that Make Decisions . . . . .</b>	<b>43</b>
	Boolean Operators . . . . .	44
	Relational Operators . . . . .	44
	Logical Operators . . . . .	45
	Relational and Logical Operator Precedence and Associativity . . . . .	46
	Comparing strings . . . . .	48
	Decision Statements . . . . .	50
	The if Statement . . . . .	50
	Exercise 4-1: Understanding if Statements . . . . .	52
	Lab 4-1: Understanding if Statements . . . . .	53
	The if else Statement . . . . .	54
	Exercise 4-2: Understanding if else Statements . . . . .	55
	Lab 4-2: Understanding if else Statements . . . . .	56
	Nested if Statements . . . . .	57
	Exercise 4-3: Understanding Nested if Statements . . . . .	58
	Lab 4-3: Understanding Nested if Statements . . . . .	59
	The switch Statement . . . . .	60
	Exercise 4-4: Using a switch Statement . . . . .	61
	Lab 4-4: Using a switch Statement . . . . .	62
	Using Decision Statements to Make Multiple Comparisons . . . . .	63
	Using AND Logic . . . . .	63
	Using OR Logic . . . . .	64
	Exercise 4-5: Making Multiple Comparisons in Decision Statements . . . . .	64
	Lab 4-5: Making Multiple Comparisons in Decision Statements . . . . .	66

<b>CHAPTER 5</b>	<b>Writing Programs Using Loops . . . . .</b>	<b>69</b>
	The Increment ( ++ ) and Decrement ( — ) Operators . . . . .	70
	Exercise 5-1: Using the Increment and Decrement Operators . . . . .	71
	Writing a while Loop in C++ . . . . .	72
	Exercise 5-2: Using a while Loop . . . . .	74
	Using a Counter to Control a Loop . . . . .	74
	Exercise 5-3: Using a Counter-Controlled while Loop . . . . .	75
	Lab 5-1: Using a Counter-Controlled while Loop . . . . .	76
	Using a Sentinel Value to Control a Loop . . . . .	76
	Exercise 5-4: Using a Sentinel Value to Control a while Loop . . . . .	79
	Lab 5-2: Using a Sentinel Value to Control a while Loop . . . . .	79
	Writing a for Loop in C++ . . . . .	80
	Exercise 5-5: Using a for Loop . . . . .	81
	Lab 5-3: Using a for Loop . . . . .	82
	Writing a do while Loop in C++ . . . . .	82
	Exercise 5-6: Using a do while Loop . . . . .	83
	Lab 5-4: Using a do while Loop . . . . .	83
	Nesting Loops . . . . .	84
	Exercise 5-7: Nesting Loops . . . . .	85
	Lab 5-5: Nesting Loops . . . . .	86
	Accumulating Totals in a Loop . . . . .	87
	Exercise 5-8: Accumulating Totals in a Loop . . . . .	88
	Lab 5-6: Accumulating Totals in a Loop . . . . .	89
	Using a Loop to Validate Input . . . . .	89
	Exercise 5-9: Validating User Input . . . . .	91
	Lab 5-7: Validating User Input . . . . .	91
 <b>CHAPTER 6</b>	 <b>Using Arrays in C++ Programs . . . . .</b>	 <b>93</b>
	Array Basics . . . . .	94
	Declaring Arrays . . . . .	94
	Initializing Arrays . . . . .	96
	Accessing Array Elements . . . . .	97
	Staying Within the Bounds of an Array . . . . .	98
	Using Constants with Arrays . . . . .	98
	Exercise 6-1: Array Basics . . . . .	99
	Lab 6-1: Array Basics . . . . .	99
	Searching an Array for an Exact Match . . . . .	100
	Exercise 6-2: Searching an Array for an Exact Match . . . . .	102
	Lab 6-2: Searching an Array for an Exact Match . . . . .	103
	Parallel Arrays . . . . .	103
	Exercise 6-3: Parallel Arrays . . . . .	106
	Lab 6-3: Parallel Arrays . . . . .	107

<b>CHAPTER 7</b>	<b>File Handling and Applications . . . . .</b>	<b>109</b>
	File Handling . . . . .	110
	Using Input and Output Classes . . . . .	110
	Opening a File for Reading . . . . .	110
	Reading Data from an Input File . . . . .	111
	Reading Data Using a Loop and EOF . . . . .	112
	Opening a File for Writing . . . . .	112
	Writing Data to an Output File . . . . .	113
	Exercise 7-1: Opening Files and Performing File Input . . . . .	114
	Lab 7-1: Opening Files and Performing File Input . . . . .	115
	Understanding Sequential Files and Control Break Logic . . . . .	116
	Exercise 7-2: Accumulating Totals in Single-Level Control	
	Break Programs . . . . .	121
	Lab 7-2: Accumulating Totals in Single-Level Control	
	Break Programs . . . . .	121
 <b>CHAPTER 8</b>	 <b>Advanced Array Techniques . . . . .</b>	 <b>125</b>
	Sorting Data . . . . .	126
	Swapping Data Values . . . . .	126
	Exercise 8-1: Swapping Values . . . . .	127
	Lab 8-1: Swapping Values . . . . .	127
	Using a Bubble Sort . . . . .	128
	The <code>main()</code> Function . . . . .	131
	The <code>fillArray()</code> Function . . . . .	132
	The <code>sortArray()</code> Function . . . . .	133
	The <code>displayArray()</code> Function . . . . .	133
	Exercise 8-2: Using a Bubble Sort . . . . .	134
	Lab 8-2: Using a Bubble Sort . . . . .	134
	Using Multidimensional Arrays . . . . .	135
	Exercise 8-3: Using Multidimensional Arrays . . . . .	138
	Lab 8-3: Using Multidimensional Arrays . . . . .	139
 <b>CHAPTER 9</b>	 <b>Advanced Modularization Techniques . . . . .</b>	 <b>141</b>
	Writing Functions with No Parameters . . . . .	142
	Exercise 9-1: Writing Functions with No Parameters . . . . .	144
	Lab 9-1: Writing Functions with No Parameters . . . . .	145
	Writing Functions that Require a Single Parameter . . . . .	145
	Exercise 9-2: Writing Functions that Require a	
	Single Parameter . . . . .	148
	Lab 9-2: Writing Functions that Require a Single Parameter . . . . .	148
	Writing Functions that Require Multiple Parameters . . . . .	149



Exercise 9-3: Writing Functions that Require Multiple Parameters . . . . .	151
Lab 9-3: Writing Functions that Require Multiple Parameters . . . . .	152
Writing Functions that Return a Value . . . . .	152
Exercise 9-4: Writing Functions that Return a Value . . . . .	155
Lab 9-4: Writing Functions that Return a Value . . . . .	155
Passing an Array and an Array Element to a Function . . . . .	156
Exercise 9-5: Passing Arrays to Functions . . . . .	159
Lab 9-5: Passing Arrays to Functions . . . . .	160
Passing Arguments by Reference and by Address . . . . .	160
Pass by Reference . . . . .	161
Pass by Address . . . . .	162
Exercise 9-6: Pass by Reference and Pass by Address . . . . .	167
Lab 9-6: Pass by Reference and Pass by Address . . . . .	168
Overloading Functions . . . . .	168
Exercise 9-7: Overloading Functions . . . . .	171
Lab 9-7: Overloading Functions . . . . .	172
Using C++ Built-in Functions . . . . .	172
Exercise 9-8: Using C++ Built-in Functions . . . . .	173
Lab 9-8: Using C++ Built-in Functions . . . . .	174

CHAPTER 10

Object-Oriented C++ . . . . .175

A Programmer-Defined Class . . . . .	176
Creating a Programmer-Defined Class . . . . .	176
Adding Attributes to a Class . . . . .	178
Adding Methods to a Class . . . . .	179
Exercise 10-1: Creating a Class in C++ . . . . .	182
Lab 10-1: Creating a Class in C++ . . . . .	183
Reusing C++ Classes . . . . .	184
Defining a Derived Class . . . . .	185
Using a Derived Class in a C++ Program . . . . .	191
Exercise 10-2: Using Inheritance to Create a Derived Class in C++ . . . . .	193
Lab 10-2: Using Inheritance to Create a Derived Class in C++ . . . . .	193

Index . . . . .195

# Preface

x

*C++ Programs to Accompany Programming Logic and Design, Eighth Edition* (also known as C++ PAL) is designed to provide students with an opportunity to write C++ programs as part of an Introductory Programming Logic course. It accompanies the student's primary text, *Programming Logic and Design, Eighth Edition*, by Joyce Farrell. This textbook assumes no programming language experience and provides the beginning programmer with a guide to writing structured programs and simple object-oriented programs using introductory elements of the popular C++ programming language. It is not intended as a textbook for a course in C++ programming. The writing is non-technical and emphasizes good programming practices. The examples do not assume mathematical background beyond high school math. Additionally, the examples illustrate one or two major points; they do not contain so many features that students become lost following irrelevant and extraneous details.

The examples in C++ PAL are often examples presented in the primary textbook, *Programming Logic and Design, Eighth Edition*. The following table shows the correlation between topics in the two books.

<b>C++ PAL, Eighth Edition</b>	<b>Programming Logic and Design, Eighth Edition</b>
Chapter 1: An Introduction to C++ and the C++ Programming Environment	Chapter 1: An Overview of Computers and Programming
Chapter 2: Variables, Constants, Operators, and Writing Programs Using Sequential Statements	Chapter 2: Elements of High-Quality Programs Chapter 3: Understanding Structure
Chapter 3: Writing Structured C++ Programs	Chapter 2: Elements of High-Quality Programs Chapter 3: Understanding Structure
Chapter 4: Writing Programs that Make Decisions	Chapter 4: Making Decisions
Chapter 5: Writing Programs Using Loops	Chapter 5: Looping
Chapter 6: Using Arrays in C++ Programs	Chapter 6: Arrays
Chapter 7: File Handling and Applications	Chapter 7: File Handling and Applications
Chapter 8: Advanced Array Techniques	Chapter 8: Advanced Data Handling Concepts
Chapter 9: Advanced Modularization Techniques	Chapter 9: Advanced Modularization Techniques
Chapter 10: Object Oriented C++	Chapter 10: Object Oriented Programming Chapter 11: More Object Oriented Programming Concepts

## Organization and Coverage

*C++ Programs to Accompany Programming Logic and Design, Eighth Edition* provides students with a review of the programming concepts they are introduced to in their primary textbook. It also shows them how to use C++ to transform their program logic and design into working programs. Chapter 1 introduces the structure of a C++ program, how to compile and run a C++ console program, and introductory object-oriented concepts. Chapter 2 discusses C++'s data types, variables, constants, arithmetic and assignment operators, and using sequential statements to write a complete C++ program. In Chapter 3, students learn how to transform pseudocode and flowcharts into C++ programs. Chapters 4 and 5 introduce students to writing C++ programs that make decisions and programs that use looping constructs. Students learn to use C++ to develop more sophisticated programs that include using arrays, control breaks, and file input and output in Chapters 6 and 7. In Chapter 8, students learn about sorting data items in an array and using multidimensional arrays. Passing parameters to functions is introduced in Chapter 9. Lastly, in Chapter 10, students use C++ to write programs that include programmer-defined classes.

This book combines text explanation of concepts and syntax along with pseudocode and actual C++ code examples to provide students with the knowledge they need to implement their logic and program designs using the C++ programming language. This book is written in a modular format and provides paper-and-pencil exercises as well as lab exercises after each major topic is introduced. The exercises provide students with experience in reading and writing C++ code as well as modifying and debugging existing code. In the labs, students are asked to complete partially prewritten C++ programs. Using partially prewritten programs allows students to focus on individual concepts rather than an entire program. The labs also allow students to see their programs execute.

C++ PAL, *Eighth Edition* is unique because:

- It is written and designed to correspond to the topics in the primary textbook, *Programming Language and Design, Eighth Edition*.
- The examples are everyday examples; no special knowledge of mathematics, accounting, or other disciplines is assumed.
- It introduces students to introductory elements of the C++ programming language rather than overwhelming beginning programmers with more detail than they are prepared to use or understand.
- Text explanations are interspersed with pseudocode from the primary book, thus reinforcing the importance of programming logic.
- Complex programs are built through the use of complete examples. Students see how an application is built from start to finish instead of studying only segments of programs.

## Features of the Text

Every chapter in this book includes the following features. These features are both conducive to learning in the classroom and enable students to learn the material at their own pace.

- **Objectives:** Each chapter begins with a list of objectives so the student knows the topics that will be presented in the chapter. In addition to providing a quick reference to topics covered, this feature provides a useful study aid.
- **Figures and illustrations:** This book has plenty of visuals, which provide the reader with a more complete learning experience, rather than one that involves simply studying text.
- **Notes:** These brief notes provide additional information—for example, a common error to watch out for.
- **Exercises:** Each section of each chapter includes meaningful paper-and-pencil exercises that allow students to practice the skills and concepts they are learning in the section.
- **Labs:** Each section of each chapter includes meaningful lab work that allows students to write and execute programs that implement their logic and program design.

## Acknowledgments

I would like to thank all of the people who helped to make this book possible. Thanks to Alyssa Pratt, Senior Content Developer, and Jim Gish, Senior Product Manager, for their help and encouragement. I am grateful to Jennifer Feltri-George, Content Project Manager, Serge Palladino, Manuscript Quality Assurance, and Anandhavalli Namachivayam of Integra Software Services, for overseeing the production of the printed book. It is a pleasure to work with so many fine people who are dedicated to producing quality instructional materials.

This book is dedicated to my mom. Her courage and determination continue to inspire me. I am grateful for such an excellent role model.

Jo Ann Smith

# Read This Before You Begin

## To the User

### Data Files

To complete most of the lab exercises, you will need data files that have been created for this book. Your instructor can provide the data files. You also can obtain the files electronically from the publisher at [www.CengageBrain.com](http://www.CengageBrain.com) (search under the ISBN for this book).

You can use a computer in your school lab or your own computer to complete the lab exercises in this book.

### Solutions

Solutions to the Exercises and Labs are provided to instructors on the Cengage Learning Web site at [sso.cengage.com](http://sso.cengage.com). The solutions are password protected.

### Using Your Own Computer

To use your own computer to complete the material in this textbook, you will need the following:

- Computer with a 1.6 GHz or faster processor
- Operating System:
  - ♦ Windows® 7 SP1 (x86 & x64)
  - ♦ Windows® 8 (x86 & x64)
  - ♦ Windows Server® 2008 R2 SP1 (x64)
  - ♦ Windows Server 2012 (x64)
- 1 GB RAM (1.5 GB if running on a virtual machine)
- 5 GB of available hard-disk space
- 5400 RPM hard drive
- DirectX 9 capable video card running at 1024 x 768 or higher display resolution

This book was written using Microsoft Windows 8 and Quality Assurance tested using Microsoft Windows 7.

## Using the C++ Compiler

xiv

To use the C++ compiler (c1) you open the Developer Command Prompt for VS2012.

To do this in Windows 8, right-click on a blank area of the **Start** screen, click **All apps**, scroll until you see the Microsoft Visual Studio 2012 area, and then click **Developer Command Prompt for VS2012**. In the next version of Windows 8, you click the down arrow on the **Start** screen, scroll until you see the Microsoft Visual Studio 2012 area, and then click **Developer Command Prompt for VS2012**.

In Windows 7, click the **Start** button, point to **All Programs**, click **Visual Studio 2012**, click **Visual Studio Tools**, and then click **Developer Command Prompt for VS2012**.

## To The Instructor

To complete some of the Exercises and Labs in this book, your students must use the data files provided with this book. These files are available from the publisher at [www.CengageBrain.com](http://www.CengageBrain.com) (search under the ISBN for this book). 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 may instruct your students to copy the files to their own computers or workstations.

## Cengage Learning 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.

EIGHTH EDITION

# C++ PROGRAMS TO ACCOMPANY PROGRAMMING LOGIC AND DESIGN

