Ninth Edition

PROBLEM SOLVING with C++

Ninth Edition

PROBLEM SOLVING with C++

Walter Savitch

UNIVERSITY OF CALIFORNIA, SAN DIEGO

CONTRIBUTOR

Kenrick Mock

UNIVERSITY OF ALASKA, ANCHORAGE

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: *Marcia Horton*
Acquisitions Editor: *Matt Goldstein*
Program Manager: *Kayla Smith-Tarbox*
Editorial Assistant: *Kelsey Loanes*
Marketing Coordinator: *Kathryn Ferranti*
Production Director: *Erin Gregg*
Managing Editor: *Scott Disanno*
Senior Operations Supervisor: *Vincent Scelta*
Operations Specialist: *Linda Sager*
Cover Designer: *Joyce Wells*
Permissions Manager: *Timothy Nicholls*
Image Permissions Manager: *Karen Sanatar*
Media Producer: *Renata Butera*
Media Project Manager: *Wanda Rockwell*
Full-Service Vendor: *Hardik Popli, Cenveo® Publisher Services*
Composition: *Cenveo Publisher Services*
Printer/Binder: *Courier/Westford*
Cover Printer: *Lehigh-Phoenix Color/Hagerstown*

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on appropriate page within text.

Library of Congress Cataloging-in-Publication Data

Savitch, Walter J., 1943-
Problem solving with C++ / Walter Savitch ; contributor, Kenrick Mock. -- Ninth edition.
pages cm
Includes index.
ISBN-13: 978-0-13-359174-3 (alkaline paper)
ISBN-10: 0-13-359174-3 (alkaline paper)
1. C++ (Computer program language) 2. Problem solving. I. Mock, Kenrick. II. Title.
QA76.73.C153S29 2014
005.13'3--dc23

2013048487

10 9 8 7 6 5 4 3 2 1—CW—15 14 13 12 11

PEARSON

www.pearsonhighered.com

ISBN 10: 0-13-359174-3
ISBN 13: 978-0-13-359174-3

Preface

This book is meant to be used in a first course in programming and computer science using the C++ language. It assumes no previous programming experience and no mathematics beyond high school algebra.

If you have used the previous edition of this book, you should read the following section that explains the changes to this ninth edition and then you can skip the rest of this preface. If you are new to this book, the rest of this preface will give you an overview of the book.

Changes to the Ninth Edition

This ninth edition presents the same programming philosophy as the eighth edition. All of the material from the eighth edition remains, but with the following enhancements:

- End-of-chapter programs are now split into Practice Programs and Programming Projects. Practice Programs require a direct application of concepts presented in the chapter and solutions are usually short. Practice Programs are appropriate for laboratory exercises. Programming Projects require additional problem solving and solutions are generally longer than Practice Programs. Programming Projects are appropriate for homework problems.
- Introduction to C++11 in the context of C++98. Examples of C++11 content includes new integer types, the auto type, raw string literals, strong enumerations, `nullptr`, ranged for loop, conversion between strings and integers, member initializers, and constructor delegation.
- Additional material on sorting, secure programming (e.g., overflow, array out of bounds), and inheritance.
- Correction of errata.
- Twenty-one new Practice Programs and ten new Programming Projects.
- Ten new VideoNotes for a total of sixty-four VideoNotes. These VideoNotes walk students through the process of both problem solving and coding to help reinforce key programming concepts. An icon appears in the margin of the book when a VideoNote is available regarding the topic covered in the text.

If you are an instructor already using the eighth edition, you can continue to teach your course almost without change.

Flexibility in Topic Ordering

This book was written to allow instructors wide latitude in reordering the material. To illustrate this flexibility, we suggest two alternative ways to order

the topics. There is no loss of continuity when the book is read in either of these ways. To ensure this continuity when you rearrange material, you may need to move sections rather than entire chapters. However, only large sections in convenient locations are moved. To help customize a particular order for any class's needs, the end of this preface contains a dependency chart, and each chapter has a "Prerequisites" section that explains what material needs to be covered before each section in that chapter.

Reordering 1: Earlier Classes

To effectively design classes, a student needs some basic tools such as control structures and function definitions. This basic material is covered in Chapters 1 through 6. After completing Chapter 6, students can begin to write their own classes. One possible reordering of chapters that allows for such early coverage of classes is the following:

Basics: Chapters 1, 2, 3, 4, 5, and 6. This material covers all control structures, function definitions, and basic file I/O. Chapter 3, which covers additional control structures, could be deferred if you wish to cover classes as early as possible.

Classes and namespaces: Chapter 10, Sections 11.1 and 11.2 of Chapter 11, and Chapter 12. This material covers defining classes, friends, overloaded operators, and namespaces.

Arrays, strings and vectors: Chapters 7 and 8

Pointers and dynamic arrays: Chapter 9

Arrays in classes: Sections 11.3 and 11.4 of Chapter 11

Inheritance: Chapter 15

Recursion: Chapter 14 (Alternately, recursion may be moved to later in the course.)

Pointers and linked lists: Chapter 13

Any subset of the following chapters may also be used:

Exception handling: Chapter 16

Templates: Chapter 17

Standard Template Library: Chapter 18

Reordering 2: Classes Slightly Later but Still Early

This version covers all control structures and the basic material on arrays before doing classes, but classes are covered later than the previous ordering and slightly earlier than the default ordering.

Basics: Chapters 1, 2, 3, 4, 5, and 6. This material covers all control structures, function definitions, and the basic file I/O.

Arrays and strings: Chapter 7, Sections 8.1 and 8.2 of Chapter 8

Classes and namespaces: Chapter 10, Sections 11.1 and 11.2 of Chapter 11, and Chapter 12. This material covers defining classes, friends, overloaded operators, and namespaces.

Pointers and dynamic arrays: Chapter 9

Arrays in classes: Sections 11.3 and 11.4 of Chapter 11

Inheritance: Chapter 15

Recursion: Chapter 14. (Alternately, recursion may be moved to later in the course.)

Vectors: Chapter 8.3

Pointers and linked lists: Chapter 13

Any subset of the following chapters may also be used:

Exception handling: Chapter 16

Templates: Chapter 17

Standard Template Library: Chapter 18

Accessibility to Students

It is not enough for a book to present the right topics in the right order. It is not even enough for it to be clear and correct when read by an instructor or other experienced programmer. The material needs to be presented in a way that is accessible to beginning students. In this introductory textbook, I have endeavored to write in a way that students find clear and friendly. Reports from the many students who have used the earlier editions of this book confirm that this style makes the material clear and often even enjoyable to students.

ANSI/ISO C++ Standard

This edition is fully compatible with compilers that meet the latest ANSI/ISO C++ standard. At the time of this writing the latest standard is C++11.

Advanced Topics

Many "advanced topics" are becoming part of a standard CS1 course. Even if they are not part of a course, it is good to have them available in the text as enrichment material. This book offers a number of advanced topics that can be integrated into a course or left as enrichment topics. It gives thorough coverage of C++ templates, inheritance (including virtual functions), exception handling, and the STL (Standard Template Library). Although this book uses libraries and teaches students the importance of libraries, it does not require any nonstandard libraries. This book uses only libraries that are provided with essentially all C++ implementations.

Dependency Chart

The dependency chart on the next page shows possible orderings of chapters and subsections. A line joining two boxes means that the upper box must be covered before the lower box. Any ordering that is consistent with this partial ordering can be read without loss of continuity. If a box contains a section number or numbers, then the box refers only to those sections and not to the entire chapter.

Summary Boxes

Each major point is summarized in a boxed section. These boxed sections are spread throughout each chapter.

Self-Test Exercises

Each chapter contains numerous Self-Test Exercises at strategic points. Complete answers for all the Self-Test Exercises are given at the end of each chapter.

VideoNotes

VideoNotes are designed for teaching students key programming concepts and techniques. These short step-by-step videos demonstrate how to solve problems from design through coding. VideoNotes allow for self-paced instruction with easy navigation including the ability to select, play, rewind, fast-forward, and stop within each VideoNote exercise.

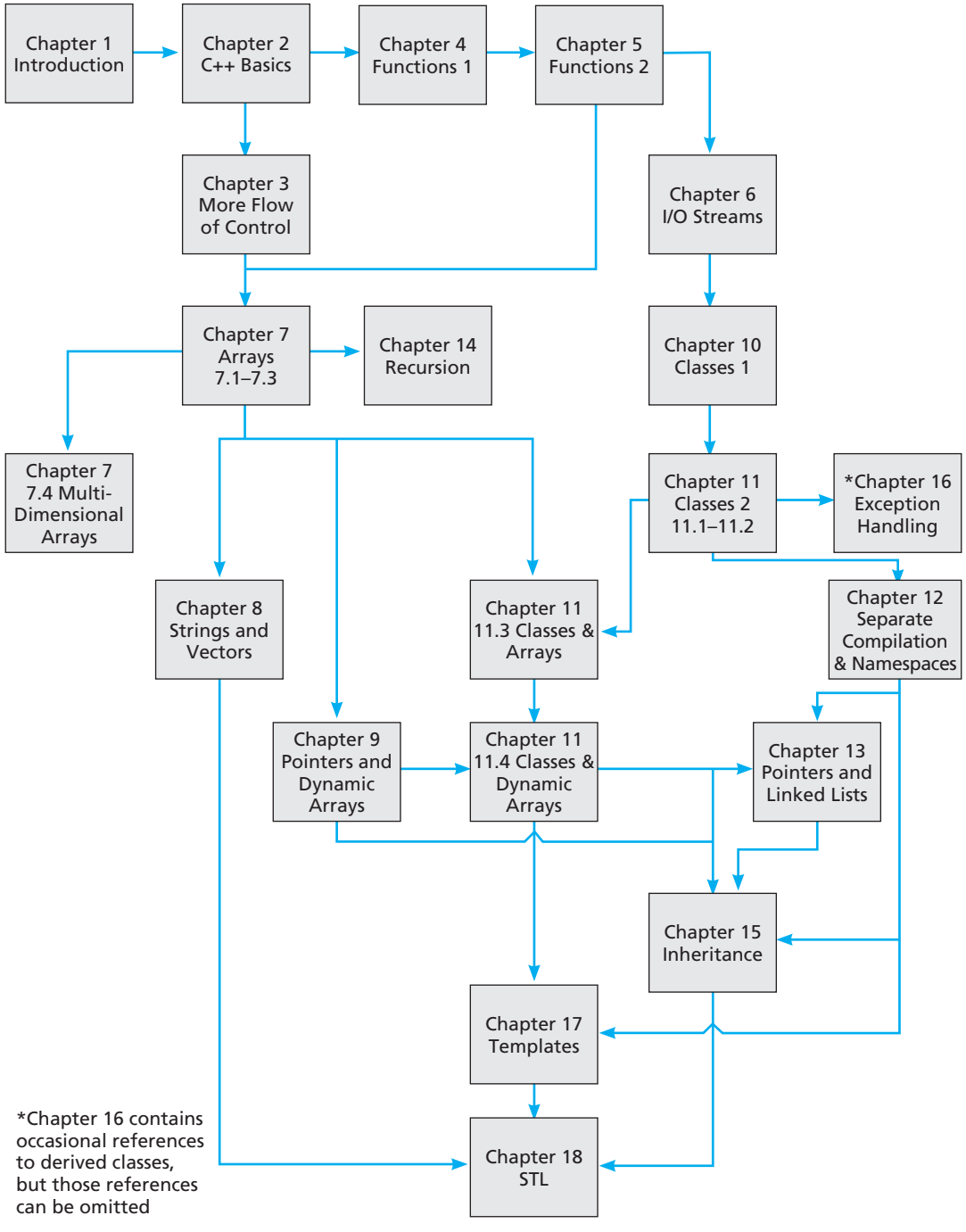
Online Practice and Assessment with MyProgrammingLab

MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students who often struggle with the basic concepts and paradigms of popular high-level programming languages.

A self-study and homework tool, a MyProgrammingLab course consists of hundreds of small practice problems organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong—and why. For instructors, a comprehensive gradebook tracks correct and incorrect answers and stores the code inputted by students for review.

MyProgrammingLab is offered to users of this book in partnership with Turing's Craft, the makers of the CodeLab interactive programming exercise system. For a full demonstration, to see feedback from instructors and students, or to get started using MyProgrammingLab in your course, visit www.myprogramminglab.com.

DISPLAY P.1 Dependency Chart



Support Material

There is support material available to all users of this book and additional material available only to qualified instructors.

Materials Available to All Users of this Book

- Source Code from the book
- PowerPoint slides
- VideoNotes

To access these materials, go to:
www.pearsonhighered.com/savitch

Resources Available to Qualified Instructors Only

Visit Pearson Education's instructor resource center at www.pearsonhighered.com/irc to access the following instructor resources:

- Instructor's Resource Guide—including chapter-by-chapter teaching hints, quiz questions with solutions, and solutions to many programming projects
- Test Bank and Test Generator
- PowerPoint Lectures—including programs and art from the text
- Lab Manual

Integrated Development Environment (IDE) Resource Kits

Instructors who adopt this text can order it for students with a kit containing five popular C++ IDEs (Microsoft® Visual Studio 2013 Express Edition, Dev C++, NetBeans, Eclipse, and CodeLite) and access to a Web site containing written and video tutorials for getting started in each IDE. For ordering information, please contact your campus Pearson Education representative.

Contact Us

Your comments, suggestions, questions, and corrections are always welcome. Please e-mail them to savitch.programming.cpp@gmail.com

Acknowledgments

Numerous individuals and groups have provided me with suggestions, discussions, and other help in preparing this textbook. Much of the first edition of this book was written while I was visiting the Computer Science Department at the University of Colorado in Boulder. The remainder of the writing on the first edition and the work on subsequent editions was done in the Computer Science and Engineering Department at the University of California, San Diego (UCSD). I am grateful to these institutions for providing a conducive environment for teaching this material and writing this book.

I extend a special thanks to all the individuals who have contributed critiques or programming projects for this or earlier editions and drafts of this book. In alphabetical order, they are: Alex Feldman, Amber Settle, Andrew Burt, Andrew Haas, Anne Marchant, Barney MacCabe, Bob Holloway, Bob Matthews, Brian R. King, Bruce Johnston, Carol Roberts, Charles Dowling, Claire Bono, Cynthia Martincic, David Feinstein, David Teague, Dennis Heckman, Donald Needham, Doug Cosman, Dung Nguyen, Edward Carr, Eitan M. Gurari, Ethan Munson, Firooz Khosraviyani, Frank Moore, Gilliean Lee, Huzefa Kagdi, James Stepleton, Jeff Roach, Jeffrey Watson, Jennifer Perkins, Jerry Weltman, Joe Faletti, Joel Cohen, John J. Westman, John Marsaglia, John Russo, Joseph Allen, Joseph D. Oldham, Jerrold Grossman, Jesse Morehouse, Karla Chaveau, Ken Rockwood, Larry Johnson, Len Garrett, Linda F. Wilson, Mal Gunasekera, Marianne Lepp, Matt Johnson, Michael Keenan, Michael Main, Michal Sramka, Naomi Shapiro, Nat Martin, Noah Aydin, Nisar Hundewale, Paul J. Kaiser, Paul Kube, Paulo Franca, Richard Borie, Scot Drysdale, Scott Strong, Sheila Foster, Steve Mahaney, Susanne Sherba, Thomas Judson, Walter A. Manrique, Wei Lian Chen, and Wojciech Komornicki.

I extend a special thanks to the many instructors who used early editions of this book. Their comments provided some of the most helpful reviewing that the book received.

Finally, I thank Kenrick Mock who implemented the changes in this edition. He had the almost impossible task of pleasing me, my editor, and his own sensibilities, and he did a superb job of it.

Walter Savitch

Brief Contents

Contents