

Preface

Welcome to *Starting Out with C++: From Control Structures through Objects, 5th edition*. In previous editions this book was called *Standard Version of Starting Out With C++*. This book is intended for use in a two-semester C++ programming sequence, or an accelerated one-semester course. Students new to programming, as well as those with prior course work in other languages, will find this text beneficial. The fundamentals of programming are covered for the novice, while the details, pitfalls, and nuances of the C++ language are explored in-depth for both the beginner and more experienced student. The book is written with clear, easy-to-understand language and it covers all the necessary topics for an introductory programming course. This text is rich in example programs that are concise, practical, and real-world oriented, ensuring that the student not only learns how to implement the features and constructs of C++, but why and when to use them.

Changes in the Fifth Edition

This book's pedagogy, organization, and clear writing style remain the same as in the previous edition. Many improvements have been made, which are summarized here:

Global Changes

- Several Programming Challenges in each chapter have been selected for inclusion in Addison-Wesley's *MyCodeMate*. *MyCodeMate* allows students to complete the Programming Challenges online, with automated assistance and feedback provided as needed. It also provides instructors with information on student progress and helps with course management.
- More Programming Challenges have been added to each chapter.
- Line numbers have been added to all of the example programs.
- More pseudocode and mini-flowchart diagrams have been added to illustrate how control structures work.
- The overall layout has been redesigned to provide a more open and accessible presentation of content.

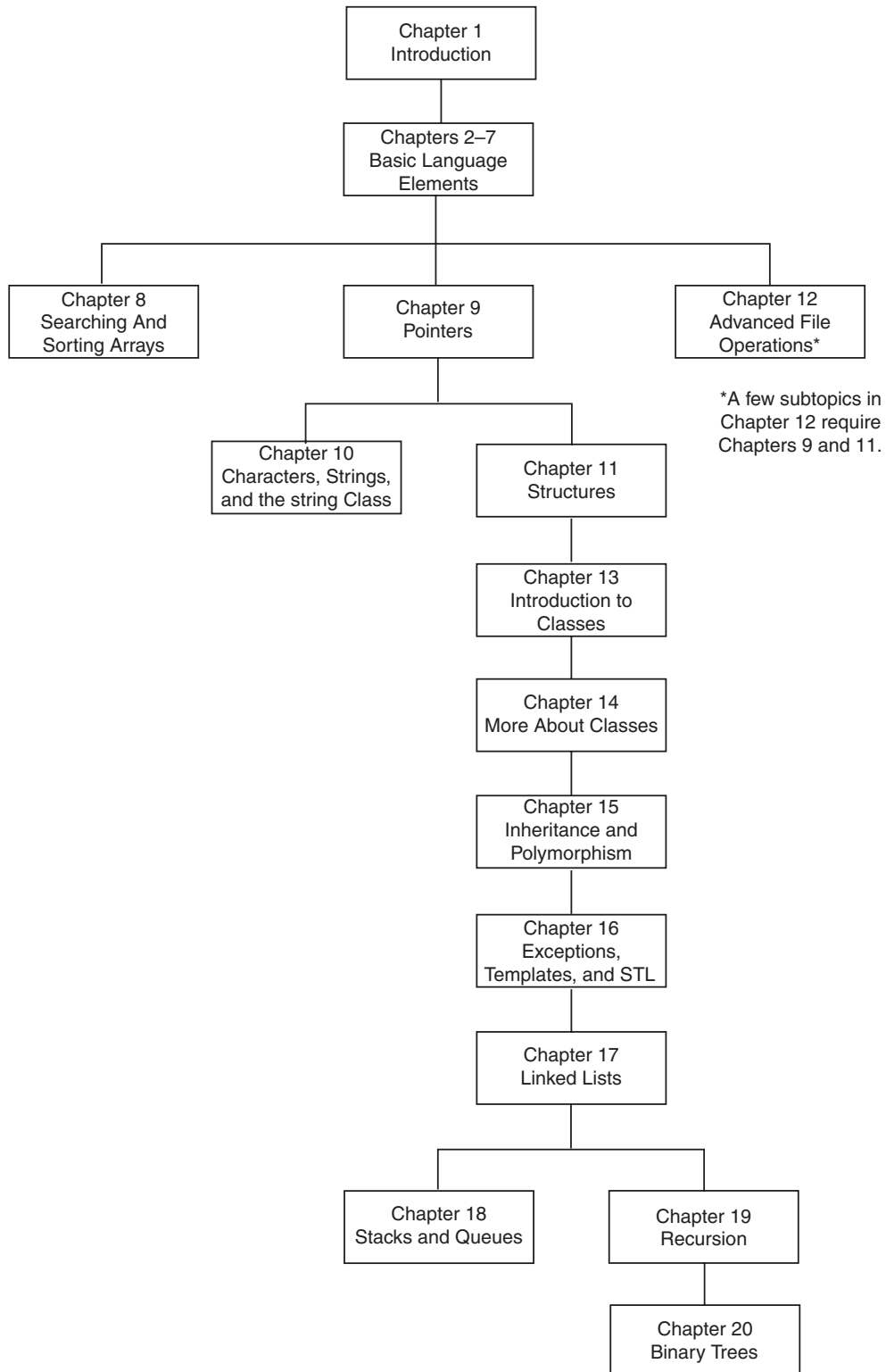
Specific Changes in each Chapter

- In Chapter 5, the `for` loop is introduced much earlier and is given greater prominence. The chapter explains in more detail how to decide which loop to use.
- Chapter 7 now discusses partially-filled arrays and how off-by-one errors can happen with arrays. The section on writing and reading arrays to and from files has been expanded.
- Chapter 9 now contrasts pointers with reference variables, explaining the similarities and differences. A discussion of pointers to `const` and constant pointers has also been added to Chapter 9.
- Chapter 11 now has an expanded discussion of dynamically allocating a `struct` variable.
- Chapter 13 has been reorganized and enhanced to focus more on object-oriented problem solving. Here are some examples of changes:
 - A more substantial introduction to object-oriented programming has been added.
 - Class member functions that do not modify class data are now declared `const`.
 - An enhanced discussion of dynamically allocating an object has been added.
 - The section on compiling multi-file programs has been enhanced.
 - A new section on using UML to model classes has been added.
 - A new section on finding the classes and their responsibilities has been added.
- Chapter 14 now has an enhanced section on aggregation, and includes a new section on class collaborations and the use of CRC cards.
- Chapter 15 has been enhanced in the following ways:
 - The discussion of base class access specification has been revised and enhanced.
 - A section on class hierarchies has been added.
 - The section on polymorphism has been completely revised. The explanations of static and dynamic binding have been enhanced to underscore the requirement for pointers or references to objects.
 - A section on virtual destructors has been added.
- New sections on recursive problem-solving and the Towers of Hanoi have been added to Chapter 19.

Organization of the Text

This text teaches C++ in a step-by-step fashion. Each chapter covers a major set of topics and builds knowledge as the student progresses through the book. Although the chapters can be easily taught in their existing sequence, some flexibility is provided. The diagram shown in Figure P-1 suggests possible sequences of instruction.

Figure P-1



Chapter 1 covers fundamental hardware, software, and programming concepts. You may choose to skip this chapter if the class has already mastered those topics. Chapters 2 through 7 cover basic C++ syntax, data types, expressions, selection structures, repetition structures, functions, and arrays. Each of these chapters builds on the previous chapter and should be covered in the order presented.

After Chapter 7 has been covered, you may proceed to Chapter 8, or jump to either Chapter 9 or Chapter 12. (If you jump to Chapter 12 at this point, you will need to postpone sections 12.7, 12.8, and 12.10 until Chapters 9 and 11 have been covered.)

After Chapter 9 has been covered, either of Chapters 10 or 11 may be covered. After Chapter 11, you may cover Chapters 13 through 17 in sequence. Next you can proceed to either Chapter 18 or Chapter 19. Finally, Chapter 20 may be covered.

This text's approach starts with a firm foundation in structured, procedural programming before delving fully into object-oriented programming and advanced data structures.

Brief Overview of Each Chapter

Chapter 1: Introduction to Computers and Programming

This chapter provides an introduction to the field of computer science and covers the fundamentals of programming, problem solving, and software design. The components of programs, such as key words, variables, operators, and punctuation are covered. The tools of the trade, such as pseudocode, flow charts, and hierarchy charts are also presented.

Chapter 2: Introduction to C++

This chapter gets the student started in C++ by introducing data types, identifiers, variable declarations, constants, comments, program output, simple arithmetic operations, and C-strings. Programming style conventions are introduced and good programming style is modeled here, as it is throughout the text. An optional section explains the difference between ANSI standard and pre-standard C++ programs.

Chapter 3: Expressions and Interactivity

In this chapter the student learns to write programs that input and handle numeric, character, and C-string data. The use of arithmetic operators and the creation of mathematical expressions are covered in greater detail, with emphasis on operator precedence. Debugging is introduced, with a section on hand tracing a program. Sections are also included on using sequential files, on simple output formatting, on data type conversion and type casting, and on using library functions that work with numbers.

Chapter 4: Making Decisions

Here the student learns about relational operators, relational expressions and how to control the flow of a program with the `if`, `if/else`, and `if/else if` statements. The conditional operator and the `switch` statement are also covered. Crucial applications of these constructs are covered, such as menu-driven programs and the validation of input.

Chapter 5: Looping

This chapter covers repetition control structures. The `while` loop, `do-while` loop, and `for` loop are taught, along with common uses for these devices. Counters, accumulators, running totals, sentinels, and other application-related topics are discussed. A section on file I/O discusses how to use a loop to read from a file until the end of the file is encountered.

Chapter 6: Functions

In this chapter the student learns how and why to modularize programs, using both `void` and value returning functions. Argument passing is covered, with emphasis on when arguments should be passed by value versus when they need to be passed by reference. Scope of variables is covered and sections are provided on local versus global variables and on static local variables. Overloaded functions are also introduced and demonstrated.

Chapter 7: Arrays

In this chapter the student learns to create and work with single and multidimensional arrays. Many examples of array processing are provided including examples illustrating how to find the sum, average, highest and lowest values in an array and how to sum the rows, columns, and all elements of a two-dimensional array. Programming techniques using parallel arrays are also demonstrated and the student is shown how to use a data file as an input source to populate an array. STL vectors are introduced and compared to arrays.

Chapter 8: Sorting and Searching Arrays

Here the student learns the basics of sorting arrays and searching for data stored in them. The chapter covers the Bubble Sort, Selection Sort, Linear Search, and Binary Search algorithms. There is also a section on sorting and searching STL vector objects.

Chapter 9: Pointers

This chapter explains how to use pointers. Pointers are compared to and contrasted with reference variables. Other topics include pointer arithmetic, initialization of pointers, relational comparison of pointers, pointers and arrays, pointers and functions, dynamic memory allocation, and more.

Chapter 10: Characters, C-strings, and the Standard `string` Class

This chapter focuses on library functions that manipulate or test characters or strings. A review of the internal storage of C-strings is given. An extensive discussion of the standard `string` class is also presented.

Chapter 11: Structured Data

The student is introduced to abstract data types and taught how to create them using structures, unions, and enumerated data types. Discussions and examples include using pointers to structures, passing structures to functions, and returning structures from functions.

Chapter 12: Advanced File and I/O Operations

This chapter covers sequential access, random access, text, and binary files. The various modes for opening files are discussed, as well as the many methods for reading and writing file contents. Advanced output formatting is also covered.

Chapter 13: Introduction to Classes

The student now shifts focus to the object-oriented paradigm. This chapter covers the fundamental concepts of classes. Member variables and functions are discussed. The student learns about private and public access specifications, and reasons to use each. The topics of constructors, overloaded constructors, and destructors are also presented. The chapter presents a section modeling classes with UML, and how to find the classes in a particular problem.

Chapter 14: More About Classes

This chapter continues the study of classes. Static members, friends, memberwise assignment, and copy constructors are discussed. The chapter also includes in-depth sections on operator overloading, object conversion, and object aggregation. There is also a section on class collaborations and the use of CRC cards.

Chapter 15: Inheritance and Polymorphism

The study of classes continues in this chapter with the subjects of inheritance, polymorphism, and virtual member functions. The topics covered include base and derived class constructors and destructors, virtual member functions, base class pointers, static and dynamic binding, multiple inheritance, and class hierarchies.

Chapter 16: Exceptions, Templates, and the Standard Template Library (STL)

The student learns to develop enhanced error trapping techniques using exceptions. Discussion then turns to function and class templates as a method for reusing code. Finally, the student is introduced to the containers, iterators, and algorithms offered by the Standard Template Library (STL).

Chapter 17: Linked Lists

This chapter introduces concepts and techniques needed to work with lists. A linked list ADT is developed and the student is taught to code operations such as creating a linked list, appending a node, traversing the list, searching for a node, inserting a node, deleting a node, and destroying a list. A linked list class template is also demonstrated.

Chapter 18: Stacks and Queues

In this chapter the student learns to create and use static and dynamic stacks and queues. The operations of stacks and queues are defined, and templates for each ADT are demonstrated.

Chapter 19: Recursion

This chapter discusses recursion and its use in problem solving. A visual trace of recursive calls is provided and recursive applications are discussed. Many recursive algorithms are presented, including recursive functions for finding factorials, finding a greatest common denominator (GCD), performing a binary search, and sorting (QuickSort). The classic Towers of Hanoi example is also presented. For students who need more challenge, there is a section on exhaustive algorithms.

Chapter 20: Binary Trees

This chapter covers the binary tree ADT, and demonstrates many binary tree operations. The student learns to traverse a tree, insert an element, delete an element, replace an element, test for an element, and destroy a tree.

Appendix A: ASCII Character Set

A list of the ASCII and Extended ASCII characters and their codes.

Appendix B: Operator Precedence and Associativity

A chart showing the C++ operators and their precedence.

The following appendices are on the accompanying Student CD

Appendix C: Introduction to Flowcharting

A brief introduction to flowcharting. This tutorial discusses sequence, selection, case, repetition, and module structures.

Appendix D: Using UML in Class Design

This appendix shows the student how to use the Unified Modeling Language to design classes. Notation for showing access specification, data types, parameters, return values, overloaded functions, composition, and inheritance are included.

Appendix E: Namespaces

This appendix explains namespaces and their purpose. Examples showing how to define a namespace and access its members are given.

Appendix F: Writing Managed C++ Code for the .NET Framework

This appendix introduces the student to the concepts surrounding managed C++ in Microsoft's .NET environment.

Appendix G: Passing Command Line Arguments

Teaches the student how to write a C++ program that accepts arguments from the command line. This appendix will be useful to students working in a command line environment, such as Unix, Linux, or the Windows MS-DOS prompt console.

Appendix H: Header File and Library Function Reference

This appendix provides a reference for the C++ library functions and header files discussed in the book.

Appendix I: Binary Numbers and Bitwise Operations

A guide to the C++ bitwise operators, as well as a tutorial on the internal storage of integers.

Appendix J: Multi-Source File Programs

Provides a tutorial on creating programs that consist of multiple source files. Function header files, class specification files, and class implementation files are discussed.

Appendix K: Stream Member Functions for Formatting

Covers stream member functions for formatting such as `setf`.

Appendix L: Installing and Using Borland C++ Builder 6

A tutorial on how to start a Borland C++ Builder 6 project, compile a program, save source files, and more.

Appendix M: Introduction to Microsoft Visual C++ 2005 Express Edition

A tutorial on how to start a project in Microsoft Visual C++ 2005 Express Edition, compile a program, save source files, and more.

Appendix N: Answers to Checkpoints

Students may test their own progress by comparing their answers to the checkpoint exercises against this appendix. The answers to all Checkpoints are included.

Appendix O: Answers to Odd-Numbered Review Questions

Another tool that students can use to gauge their progress.

Features of the Text

Concept Statements	Each major section of the text starts with a concept statement. This statement summarizes the ideas of the section.
Example Programs	The text has over 300 complete example programs, each designed to highlight the topic currently being studied. In most cases, these are practical, real-world examples. Source code for these programs is provided so that students can run the programs themselves.
Program Output	After each example program there is a sample of its screen output. This immediately shows the student how the program should function.
Checkpoints	Checkpoints are questions placed throughout each chapter as a self-test study aid. Answers for all Checkpoint questions are provided on the student CD so students can check how well they have learned a new topic.
Notes	Notes appear at appropriate places throughout the text. They are short explanations of interesting or often misunderstood points relevant to the topic at hand.
Warnings	Warnings are notes that caution the student about certain C++ features, programming techniques, or practices that can lead to malfunctioning programs or lost data.
Case Studies	Case studies that simulate real-world applications appear in many chapters throughout the text, with complete code provided for each one on the student CD. These case studies are designed to highlight the major topics of the chapter in which they appear.
Review Questions and Exercises	Each chapter presents a thorough and diverse set of review questions, such as fill-in-the-blank and short answer, that check the student's mastery of the basic material presented in the chapter. These are followed by exercises requiring problem solving and analysis, such as the <i>Algorithm Workbench</i> , <i>Predict the Output</i> , and <i>Find the Errors</i> sections. Answers to the odd numbered review questions and review exercises are provided on the student CD.
Programming Challenges	Each chapter offers a pool of programming exercises designed to solidify the student's knowledge of the topics currently being studied. In most cases the assignments present real-world problems to be solved. When applicable, these exercises include input validation rules.

Group Projects

There are several group programming projects throughout the text, intended to be constructed by a team of students. One student might build the program's user interface, while another student writes the mathematical code, and another designs and implements a class the program uses. This process is similar to the way many professional programs are written and encourages team work within the classroom.

**Software
Development
Project:
Serendipity
Booksellers**

This is an on-going project that instructors can optionally assign to teams of students. It systematically develops a "real-world" software package: a point-of-sale program for the fictitious Serendipity Booksellers organization. The Serendipity assignment for each chapter adds more functionality to the software, using constructs and techniques covered in that chapter. When complete, the program will act as a cash register, manage an inventory database, and produce a variety of reports.

Supplements

Student CD

This CD includes:

- Borland C++ Builder 6.0, Personal Edition
- Answers to all Checkpoint questions (Appendix N)
- Answers to all odd-numbered Review Questions and Exercises (Appendix O)
- Complete source code for every program included in the book
- Additional case studies, complete with source code
- A full set of appendices (including several tutorials) that accompany the book

If a CD did not come with your book or you can't locate your CD, you can access most of these items at <http://www.aw.com/cssupport/>

Other CDs Upon Request

Professors should contact their campus Addison-Wesley representative for the specific ISBN to order this book packaged with another CD.

- Microsoft Visual C++ 2005 Express Edition

MyCodeMate—Your Own T.A. Just a Click Away

Addison-Wesley's *MyCodeMate* is a book-specific Web resource that provides tutorial help and evaluation of student programs. Example programs throughout the book and selected Programming Challenges from every chapter have been integrated into *MyCodeMate*. Using this tool, a student is able to write and compile programs from any computer with Internet access, and receive guidance and feedback on how to proceed and on how to address compiler error messages. Instructors can track each student's progress on Programming Challenges from the text or can develop projects of their own. **A complimentary subscription to *MyCodeMate* is offered when the access code is ordered in a package with a new copy of this text.** Subscriptions can also be purchased online. For more information visit <http://www.mycodemate.com>, or contact your campus Addison-Wesley representative.

Instructor Resources

The following supplements are available to qualified instructors only:

- Answers to all Review Questions in the text
- Solutions for all Programming Challenges in the text
- PowerPoint presentation slides for every chapter
- Computerized test bank
- Answers to all Student Lab Manual questions
- Solutions for all Student Lab Manual programs

Visit the Addison-Wesley Instructor Resource Center (<http://www.aw.com/irc>) or send an email to computing@aw.com for information on how to access them.

Additional Supplements

The following supplementary material is also available for this textbook. Contact your campus Addison-Wesley representative for more information.

- *Lab Manual to Accompany Starting Out with C++: Control Structures through Objects, 5th Edition*, co-authored by Dean Defino and Michael Bardzell

Textbook Web site

A Web site for the *Starting Out with C++* series of books is located at the following URL: <http://www.aw.com/gaddisbooks>

Which Gaddis C++ book is right for you?

The Starting Out with C++ Series includes three books, one of which is sure to fit your course:

- *Starting Out with C++: From Control Structures through Objects* (formerly called the “Standard Version”);
- *Starting Out with C++: Early Objects* (formerly called the “Alternate Version”);
- *Starting Out with C++: Brief Version*.

The following chart will help you determine which book is right for your course.

■ FROM CONTROL STRUCTURES THROUGH OBJECTS ■ BRIEF VERSION	■ EARLY OBJECTS
<p>LATE INTRODUCTION OF OBJECTS Classes are introduced in Chapter 13 of the standard text and Chapter 11 of the brief text, after control structures, functions, arrays, and pointers. Advanced OOP topics, such as inheritance and polymorphism, are covered in the following two chapters.</p> <p>USE OF C-STRINGS Null-terminated C-strings are used throughout, with the C++ <code>string</code> class covered briefly.</p> <p>INTRODUCTION OF DATA STRUCTURES AND RECURSION Linked lists, stacks and queues, and binary trees are introduced in the final chapters of the standard text. Recursion is covered after stacks and queues, but before binary trees. These topics are not covered in the brief text, though it does have appendices dealing with linked lists and recursion.</p>	<p>EARLIER INTRODUCTION OF OBJECTS Classes are introduced in Chapter 7, after control structures and functions, but before arrays and pointers. Their use is then integrated into the remainder of the text. Advanced OOP topics, such as inheritance and polymorphism, are covered in Chapters 11 and 15.</p> <p>USE OF <code>string</code> OBJECTS Standard library <code>string</code> class objects are used throughout, with C-strings covered briefly.</p> <p>INTRODUCTION OF DATA STRUCTURES AND RECURSION Linked lists, stacks and queues, and binary trees are introduced in the final chapters of the text, after the chapter on recursion.</p>

Acknowledgments

There have been many helping hands in the development and publication of this text. We would like to thank the following faculty reviewers for their helpful suggestions and expertise.

Reviewers of the Fifth Edition

Vicki Allan
Utah State University

Mary Astone
Troy University

Judy Etchison
Southern Methodist University

Ann Ford
Florida State University

Sheila Lancaster
Gadsden State Community College

Farshad Ravanshad
Bergen Community College

Jason Schatz
City College of San Francisco

William Slater
Collin County Community College

Dale Suggs
Campbell University

Peter van der Goes
Rose State College

Aurelia Williams
Norfolk State University

Reviewers of Previous Editions

Ahmad Abuhejleh
University of Wisconsin, River Falls

David Akins
El Camino College

Vicki Allan
Utah State University

Steve Allan
Utah State University

Karen M. Arlien
Bismark State College

Ijaz A. Awan
Savannah State University

Robert Baird
Salt Lake Community College

Don Biggerstaff
Fayetteville Technical Community College

Michael Bolton
Northeastern Oklahoma State University

Jennifer Li
Oblone College

Norman H. Liebling
San Jacinto College

Zhu-qu Lu
University of Maine, Presque Isle

Heidar Malki
University of Houston

Debbie Mathews
J. Sargeant Reynolds

Rick Matzen
Northeastern State University

Robert McDonald
East Stroudsburg University

James McGuffee
Austin Community College

Dean Mellas
Cerritos College

- Bill Brown
Pikes Peak Community College
- Charles Cadenhead
Richland Community College
- Randall Campbell
Morningside College
- Wayne Caruolo
Red Rocks Community College
- Cathi Chambley-Miller
Aiken Technical College
- C.C. Chao
Jacksonville State University
- Joseph Chao
Bowling Green State University
- Royce Curtis
Western Wisconsin Technical College
- Joseph DeLibero
Arizona State University
- Jeanne Douglas
University of Vermont
- Michael Dowell
Augusta State U
- Judy Etchison
University of Texas at Dallas
- Dennis Fairclough
Utah Valley State College
- Richard Flint
North Central College
- James Gifford
University of Wisconsin, Stevens Point
- Leon Gleiberman
Touro College
- Ranette Halverson, Ph.D.
Midwestern State University
- Carol Hannahs
University of Kentucky
- Dennis Heckman
Portland Community College
- Lisa Milkowski
Milwaukee School of Engineering
- Marguerite Nedreberg
Youngstown State University
- Lynne O'Hanlon
Los Angeles Pierce College
- Frank Paiano
Southwestern Community College
- Theresa Park
Texas State Technical College
- Mark Parker
Shoreline Community College
- Tino Posillico
SUNY Farmingdale
- Frederick Pratter
Eastern Oregon University
- Susan L. Quick
Penn State University
- Alberto Ramon
Diablo Valley College
- Bazlur Rasheed
Sault College of Applied Arts and Technology
- Dolly Samson
Weber State University
- Ruth Sapir
SUNY Farmingdale
- Dr. Sung Shin
South Dakota State University
- Bari Siddique
University of Texas at Brownsville
- Shep Smithline
University of Minnesota
- Caroline St. Claire
North Central College
- Kirk Stephens
Southwestern Community College
- Cherie Stevens
South Florida Community College

Ric Heishman <i>Northern Virginia Community College</i>	Dale Suggs <i>Campbell University</i>
Michael Hennessy <i>University of Oregon</i>	Mark Swanson <i>Red Wing Technical College</i>
Ilga Higbee <i>Black Hawk College</i>	Martha Tillman <i>College of San Mateo</i>
Patricia Hines <i>Brookdale Community College</i>	Ralph Tomlinson <i>Iowa State University</i>
Mike Holland <i>Northern Virginia Community College</i>	David Topham <i>Oblone College</i>
Mary Hovik <i>Lehigh Carbon Community College</i>	Rober Tureman <i>Paul D. Camp Community College</i>
Richard Hull <i>Lenoir-Rhyne College</i>	Arisa K. Ude <i>Richland College</i>
Chris Kardaras <i>North Central College</i>	Stewart Venit <i>California State University, Los Angeles</i>
Willard Keeling <i>Blue Ridge Community College</i>	Judy Walters <i>North Central College</i>
A.J. Krygeris <i>Houston Community College</i>	John H. Whipple <i>Northampton Community College</i>
Ray Larson <i>Inver Hills Community College</i>	Vida Winans <i>Illinois Institute of Technology</i>

First, I would like to thank my family for their many years of support, and my students at Haywood Community College for inspiring me to write student-friendly books. I owe a debt of gratitude to everyone at Addison-Wesley who is part of the editorial, production, and marketing team for this book. I am very fortunate to have Michael Hirsch as my editor, guiding me through all phases of development and production. I am also fortunate to work with Michelle Brown, Patty Mahtani, Lindsey Triebel, Bethany Tidd, Joyce Wells, Mario Rodriguez, Stephen Adams, and Holly McLean-Aldis. This book would not be possible without your tireless work. Finally, I want to thank my long-time friend Richard Jones for his mentorship. You are all great people to work with!

About the Author

Tony Gaddis is the principal author of the “Starting Out with” series of textbooks. Tony teaches computer science courses at Haywood Community College in North Carolina. He is a highly acclaimed instructor who was previously selected as the North Carolina Community College “Teacher of the Year,” and has received the Teaching Excellence award from the National Institute for Staff and Organizational Development. Besides C++ books, the “Starting Out with” series includes introductory books using the Java™ programming language, Microsoft® Visual Basic® .NET, and Microsoft® C#®, all published by Addison-Wesley.