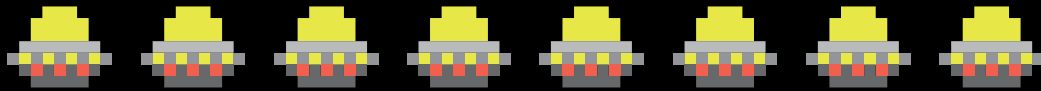
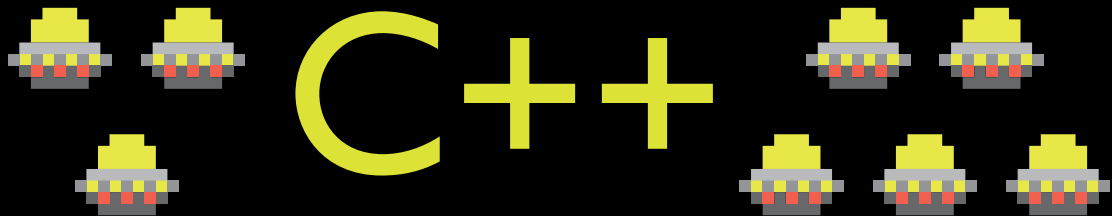


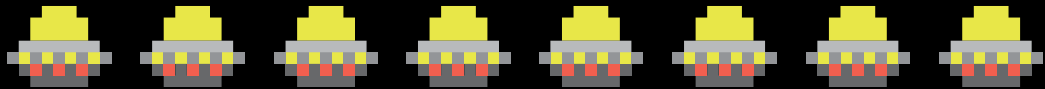
GLOBAL
EDITION



 Starting Out with  

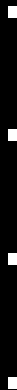


From Control Structures through Objects



NINTH EDITION

Tony Gaddis



Digital Resources for Students

Your new textbook provides 12-month access to digital resources that may include VideoNotes (step-by-step video tutorials on programming concepts), source code, web chapters, quizzes, and more. Refer to the preface in the textbook for a detailed list of resources.

Follow the instructions below to register for the Companion Website for Tony Gaddis' *Starting Out with C++ from Control Structures through Objects*, Ninth Edition, Global Edition.

1. Go to www.pearsonglobaleditions.com/gaddis
2. Enter the title of your textbook or browse by author name.
3. Click Companion Website.
4. Click Register and follow the on-screen instructions to create a login name and password.

ISSGDS-SPADA-BAUTH-SKEAN-FIERI-BALES

Use the login name and password you created during registration to start using the online resources that accompany your textbook.

IMPORTANT:

This prepaid subscription does not include access to MyProgrammingLab, which is available at www.myprogramminglab.com for purchase.

This access code can only be used once. This subscription is valid for 12 months upon activation and is not transferable.

For technical support go to <https://support.pearson.com/getsupport>

This page intentionally left blank

STARTING OUT WITH

C++

From Control Structures
through Objects

NINTH EDITION

GLOBAL EDITION

This page intentionally left blank

STARTING OUT WITH

C++

From Control Structures
through Objects

NINTH EDITION

GLOBAL EDITION

Tony Gaddis

Haywood Community College



330 Hudson Street, New York, NY 10013

Senior Vice President Courseware Portfolio
Management: Marcia J. Horton
Director, Portfolio Management: Engineering,
Computer Science & Global Editions: Julian Partridge
Portfolio Manager: Matt Goldstein
Portfolio Management Assistant: Kristy Alaura
Assistant Acquisitions Editor, Global Edition:
Aditee Agarwal
Field Marketing Manager: Demetrius Hall
Product Marketing Manager: Yvonne Vannatta

Managing Producer, ECS and Math: Scott Disanno
Content Producer: Sandra L. Rodriguez
Senior Project Editor, Global Edition: K.K.
Neelakantan
Senior Manufacturing Controller, Global Edition:
Caterina Pellegrino
Manager, Media Production, Global Edition:
Vikram Kumar
Cover Designer: Lumina Datamatics, Inc.
Cover Photo: dmitriylo/Shutterstock

Credits and acknowledgments borrowed from other sources and reproduced, with permission, appear on the Credits page in the endmatter of this textbook.

Pearson Education Limited
KAO Two
KAO Park
Harlow
CM17 9NA
United Kingdom

and Associated Companies throughout the world

Visit us on the World Wide Web at: www.pearsonglobaleditions.com

© Pearson Education Limited 2019

The right of Tony Gaddis to be identified as the author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

Authorized adaptation from the United States edition, entitled Starting Out with C++: From Control Structures through Objects, 9th Edition, ISBN 978-0-13-449837-9 by Tony Gaddis, published by Pearson Education © 2018.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6–10 Kirby Street, London EC1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

ISBN 10: 1-292-22233-6

ISBN 13: 978-1-292-22233-2

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

Typeset by iEnergizer Aptara®, Ltd

Printed and bound by Vivar in Malaysia

Contents at a Glance

	Preface	17
CHAPTER 1	Introduction to Computers and Programming	33
CHAPTER 2	Introduction to C++	59
CHAPTER 3	Expressions and Interactivity	117
CHAPTER 4	Making Decisions	183
CHAPTER 5	Loops and Files	263
CHAPTER 6	Functions	337
CHAPTER 7	Arrays and Vectors	413
CHAPTER 8	Searching and Sorting Arrays	495
CHAPTER 9	Pointers	535
CHAPTER 10	Characters, C-Strings, and More about the string Class	559
CHAPTER 11	Structured Data	645
CHAPTER 12	Advanced File Operations	697
CHAPTER 13	Introduction to Classes	751
CHAPTER 14	More about Classes	849
CHAPTER 15	Inheritance, Polymorphism, and Virtual Functions	939
CHAPTER 16	Exceptions and Templates	1021
CHAPTER 17	The Standard Template Library	1061
CHAPTER 18	Linked Lists	1155
CHAPTER 19	Stacks and Queues	1197
CHAPTER 20	Recursion	1255
CHAPTER 21	Binary Trees	1289
	Appendix A: The ASCII Character Set	1319
	Appendix B: Operator Precedence and Associativity	1321

Quick References 1323

Index 1325

Credit 1343

Online The following appendices are available at www.pearsonglobaleditions.com/gaddis.

Appendix C: Introduction to Flowcharting

Appendix D: Using UML in Class Design

Appendix E: Namespaces

Appendix F: Passing Command Line Arguments

Appendix G: Binary Numbers and Bitwise Operations

Appendix H: STL Algorithms

Appendix I: Multi-Source File Programs

Appendix J: Stream Member Functions for Formatting

Appendix K: Unions

Appendix L: Answers to Checkpoints

Appendix M: Answers to Odd Numbered Review Questions

Case Study 1: String Manipulation

Case Study 2: High Adventure Travel Agency—Part 1

Case Study 3: Loan Amortization

Case Study 4: Creating a String Class

Case Study 5: High Adventure Travel Agency—Part 2

Case Study 6: High Adventure Travel Agency—Part 3

Case Study 7: Intersection of Sets

Case Study 8: Sales Commission

Contents

Preface 17

CHAPTER 1 Introduction to Computers and Programming 33

- 1.1 Why Program? 33
- 1.2 Computer Systems: Hardware and Software 34
- 1.3 Programs and Programming Languages 40
- 1.4 What Is a Program Made of? 46
- 1.5 Input, Processing, and Output 49
- 1.6 The Programming Process 50
- 1.7 Procedural and Object-Oriented Programming 54
- Review Questions and Exercises 56*

CHAPTER 2 Introduction to C++ 59

- 2.1 The Parts of a C++ Program 59
- 2.2 The `cout` Object 63
- 2.3 The `#include` Directive 68
- 2.4 Variables, Literals, and Assignment Statements 70
- 2.5 Identifiers 74
- 2.6 Integer Data Types 75
- 2.7 The `char` Data Type 81
- 2.8 The C++ `string` Class 85
- 2.9 Floating-Point Data Types 87
- 2.10 The `bool` Data Type 90
- 2.11 Determining the Size of a Data Type 91
- 2.12 More about Variable Assignments and Initialization 92
- 2.13 Scope 94
- 2.14 Arithmetic Operators 95
- 2.15 Comments 103
- 2.16 Named Constants 105
- 2.17 Programming Style 107
- Review Questions and Exercises 109*
- Programming Challenges 113*

CHAPTER 3 Expressions and Interactivity 117

- 3.1 The `cin` Object 117
- 3.2 Mathematical Expressions 123
- 3.3 When You Mix Apples and Oranges: Type Conversion 132
- 3.4 Overflow and Underflow 134
- 3.5 Type Casting 135
- 3.6 Multiple Assignment and Combined Assignment 138
- 3.7 Formatting Output 142
- 3.8 Working with Characters and `string` Objects 152
- 3.9 More Mathematical Library Functions 158
- 3.10 Focus on Debugging: Hand Tracing a Program 164
- 3.11 Focus on Problem Solving: A Case Study 166
- Review Questions and Exercises* 170
- Programming Challenges* 176

CHAPTER 4 Making Decisions 183

- 4.1 Relational Operators 183
- 4.2 The `if` Statement 188
- 4.3 Expanding the `if` Statement 196
- 4.4 The `if/else` Statement 200
- 4.5 Nested `if` Statements 203
- 4.6 The `if/else if` Statement 210
- 4.7 Flags 215
- 4.8 Logical Operators 216
- 4.9 Checking Numeric Ranges with Logical Operators 223
- 4.10 Menus 224
- 4.11 Focus on Software Engineering: Validating User Input 227
- 4.12 Comparing Characters and Strings 229
- 4.13 The Conditional Operator 233
- 4.14 The `switch` Statement 236
- 4.15 More about Blocks and Variable Scope 245
- Review Questions and Exercises* 248
- Programming Challenges* 254

CHAPTER 5 Loops and Files 263

- 5.1 The Increment and Decrement Operators 263
- 5.2 Introduction to Loops: The `while` Loop 268
- 5.3 Using the `while` Loop for Input Validation 275
- 5.4 Counters 277
- 5.5 The `do-while` Loop 278
- 5.6 The `for` Loop 283
- 5.7 Keeping a Running Total 293
- 5.8 Sentinels 296
- 5.9 Focus on Software Engineering: Deciding Which Loop to Use 297
- 5.10 Nested Loops 298
- 5.11 Using Files for Data Storage 301
- 5.12 Optional Topics: Breaking and Continuing a Loop 320
- Review Questions and Exercises* 324
- Programming Challenges* 329

CHAPTER 6 Functions 337

- 6.1 Focus on Software Engineering: Modular Programming 337
- 6.2 Defining and Calling Functions 338
- 6.3 Function Prototypes 347
- 6.4 Sending Data into a Function 349
- 6.5 Passing Data by Value 354
- 6.6 Focus on Software Engineering: Using Functions in a
Menu-Driven Program 356
- 6.7 The return Statement 360
- 6.8 Returning a Value from a Function 362
- 6.9 Returning a Boolean Value 370
- 6.10 Local and Global Variables 372
- 6.11 Static Local Variables 380
- 6.12 Default Arguments 383
- 6.13 Using Reference Variables as Parameters 386
- 6.14 Overloading Functions 392
- 6.15 The `exit()` Function 396
- 6.16 Stubs and Drivers 399
- Review Questions and Exercises* 401
- Programming Challenges* 404

CHAPTER 7 Arrays and Vectors 413

- 7.1 Arrays Hold Multiple Values 413
- 7.2 Accessing Array Elements 415
- 7.3 No Bounds Checking in C++ 427
- 7.4 The Range-Based for Loop 430
- 7.5 Processing Array Contents 434
- 7.6 Focus on Software Engineering: Using Parallel Arrays 442
- 7.7 Arrays as Function Arguments 445
- 7.8 Two-Dimensional Arrays 456
- 7.9 Arrays with Three or More Dimensions 463
- 7.10 Focus on Problem Solving and Program Design: A Case Study 465
- 7.11 Introduction to the STL vector 467
- Review Questions and Exercises* 481
- Programming Challenges* 486

CHAPTER 8 Searching and Sorting Arrays 495

- 8.1 Focus on Software Engineering: Introduction to Search Algorithms 495
- 8.2 Focus on Problem Solving and Program Design: A Case Study 501
- 8.3 Focus on Software Engineering: Introduction to Sorting Algorithms 508
- 8.4 Focus on Problem Solving and Program Design: A Case Study 518
- 8.5 Sorting and Searching vectors 527
- Review Questions and Exercises* 530
- Programming Challenges* 531

CHAPTER 9 Pointers 535

- 9.1 Getting the Address of a Variable 535
- 9.2 Pointer Variables 537
- 9.3 The Relationship between Arrays and Pointers 544

- 9.4 Pointer Arithmetic 548
- 9.5 Initializing Pointers 550
- 9.6 Comparing Pointers 551
- 9.7 Pointers as Function Parameters 553
- 9.8 Dynamic Memory Allocation 562
- 9.9 Returning Pointers from Functions 566
- 9.10 Using Smart Pointers to Avoid Memory Leaks 573
- 9.11 Focus on Problem Solving and Program Design: A Case Study 576
- Review Questions and Exercises* 582
- Programming Challenges* 585

CHAPTER 10 Characters, C-Strings, and More about the string Class 589

- 10.1 Character Testing 589
- 10.2 Character Case Conversion 593
- 10.3 C-Strings 596
- 10.4 Library Functions for Working with C-Strings 600
- 10.5 String/Numeric Conversion Functions 611
- 10.6 Focus on Software Engineering: Writing Your Own C-String-Handling Functions 617
- 10.7 More about the C++ string Class 623
- 10.8 Focus on Problem Solving and Program Design: A Case Study 635
- Review Questions and Exercises* 636
- Programming Challenges* 639

CHAPTER 11 Structured Data 645

- 11.1 Abstract Data Types 645
- 11.2 Structures 647
- 11.3 Accessing Structure Members 650
- 11.4 Initializing a Structure 654
- 11.5 Arrays of Structures 657
- 11.6 Focus on Software Engineering: Nested Structures 659
- 11.7 Structures as Function Arguments 663
- 11.8 Returning a Structure from a Function 666
- 11.9 Pointers to Structures 669
- 11.10 Focus on Software Engineering: When to Use ., When to Use ->, and When to Use * 672
- 11.11 Enumerated Data Types 674
- Review Questions and Exercises* 685
- Programming Challenges* 691

CHAPTER 12 Advanced File Operations 697

- 12.1 File Operations 697
- 12.2 File Output Formatting 703
- 12.3 Passing File Stream Objects to Functions 705
- 12.4 More Detailed Error Testing 707
- 12.5 Member Functions for Reading and Writing Files 710
- 12.6 Focus on Software Engineering: Working with Multiple Files 718
- 12.7 Binary Files 720
- 12.8 Creating Records with Structures 725

- 12.9 Random-Access Files 729
- 12.10 Opening a File for Both Input and Output 737
 - Review Questions and Exercises* 742
 - Programming Challenges* 745

CHAPTER 13 Introduction to Classes 751

- 13.1 Procedural and Object-Oriented Programming 751
- 13.2 Introduction to Classes 758
- 13.3 Defining an Instance of a Class 763
- 13.4 Why Have Private Members? 776
- 13.5 Focus on Software Engineering: Separating Class Specification from Implementation 777
- 13.6 Inline Member Functions 783
- 13.7 Constructors 786
- 13.8 Passing Arguments to Constructors 791
- 13.9 Destructors 799
- 13.10 Overloading Constructors 803
- 13.11 Private Member Functions 807
- 13.12 Arrays of Objects 809
- 13.13 Focus on Problem Solving and Program Design: An OOP Case Study 813
- 13.14 Focus on Object-Oriented Programming: Simulating Dice with Objects 820
- 13.15 Focus on Object-Oriented Design: The Unified Modeling Language (UML) 824
- 13.16 Focus on Object-Oriented Design: Finding the Classes and Their Responsibilities 826
 - Review Questions and Exercises* 835
 - Programming Challenges* 840

CHAPTER 14 More about Classes 849

- 14.1 Instance and Static Members 849
- 14.2 Friends of Classes 857
- 14.3 Memberwise Assignment 862
- 14.4 Copy Constructors 863
- 14.5 Operator Overloading 869
- 14.6 Object Conversion 896
- 14.7 Aggregation 898
- 14.8 Focus on Object-Oriented Design: Class Collaborations 903
- 14.9 Focus on Object-Oriented Programming: Simulating the Game of Cho-Han 908
- 14.10 Rvalue References and Move Semantics 918
 - Review Questions and Exercises* 927
 - Programming Challenges* 932

CHAPTER 15 Inheritance, Polymorphism, and Virtual Functions 939

- 15.1 What Is Inheritance? 939
- 15.2 Protected Members and Class Access 948
- 15.3 Constructors and Destructors in Base and Derived Classes 954
- 15.4 Redefining Base Class Functions 968
- 15.5 Class Hierarchies 973
- 15.6 Polymorphism and Virtual Member Functions 979
- 15.7 Abstract Base Classes and Pure Virtual Functions 995

- 15.8 Multiple Inheritance 1002
- Review Questions and Exercises* 1009
- Programming Challenges* 1013

CHAPTER 16 Exceptions and Templates 1021

- 16.1 Exceptions 1021
- 16.2 Function Templates 1040
- 16.3 Focus on Software Engineering: Where to Start When Defining Templates 1046
- 16.4 Class Templates 1046
- Review Questions and Exercises* 1056
- Programming Challenges* 1058

CHAPTER 17 The Standard Template Library 1061

- 17.1 Introduction to the Standard Template Library 1061
- 17.2 STL Container and Iterator Fundamentals 1061
- 17.3 The `vector` Class 1072
- 17.4 The `map`, `multimap`, and `unordered_map` Classes 1086
- 17.5 The `set`, `multiset`, and `unordered_set` Classes 1111
- 17.6 Algorithms 1118
- 17.7 Introduction to Function Objects and Lambda Expressions 1139
- Review Questions and Exercises* 1146
- Programming Challenges* 1150

CHAPTER 18 Linked Lists 1155

- 18.1 Introduction to the Linked List ADT 1155
- 18.2 Linked List Operations 1157
- 18.3 A Linked List Template 1173
- 18.4 Variations of the Linked List 1185
- 18.5 The STL `list` and `forward_list` Containers 1186
- Review Questions and Exercises* 1190
- Programming Challenges* 1193

CHAPTER 19 Stacks and Queues 1197

- 19.1 Introduction to the Stack ADT 1197
- 19.2 Dynamic Stacks 1214
- 19.3 The STL `stack` Container 1225
- 19.4 Introduction to the Queue ADT 1227
- 19.5 Dynamic Queues 1239
- 19.6 The STL `deque` and `queue` Containers 1246
- Review Questions and Exercises* 1249
- Programming Challenges* 1251

CHAPTER 20 Recursion 1255

- 20.1 Introduction to Recursion 1255
- 20.2 Solving Problems with Recursion 1259
- 20.3 Focus on Problem Solving and Program Design: The Recursive `gcd` Function 1267
- 20.4 Focus on Problem Solving and Program Design: Solving Recursively Defined Problems 1268

- 20.5 Focus on Problem Solving and Program Design: Recursive Linked List Operations 1269
- 20.6 Focus on Problem Solving and Program Design: A Recursive Binary Search Function 1273
- 20.7 The Towers of Hanoi 1275
- 20.8 Focus on Problem Solving and Program Design: The QuickSort Algorithm 1278
- 20.9 Exhaustive Algorithms 1282
- 20.10 Focus on Software Engineering: Recursion versus Iteration 1285
- Review Questions and Exercises* 1285
- Programming Challenges* 1287

CHAPTER 21 Binary Trees 1289

- 21.1 Definition and Applications of Binary Trees 1289
- 21.2 Binary Search Tree Operations 1292
- 21.3 Template Considerations for Binary Search Trees 1309
- Review Questions and Exercises* 1315
- Programming Challenges* 1316

Appendix A: The ASCII Character Set 1319

Appendix B: Operator Precedence and Associativity 1321

Quick References 1323

Index 1325

Credit 1343

Online The following appendices are available at www.pearsonglobaleditions.com/gaddis.

Appendix C: Introduction to Flowcharting

Appendix D: Using UML in Class Design

Appendix E: Namespaces

Appendix F: Passing Command Line Arguments

Appendix G: Binary Numbers and Bitwise Operations

Appendix H: STL Algorithms

Appendix I: Multi-Source File Programs

Appendix J: Stream Member Functions for Formatting

Appendix K: Unions

Appendix L: Answers to Checkpoints

Appendix M: Answers to Odd Numbered Review Questions

Case Study 1: String Manipulation

Case Study 2: High Adventure Travel Agency—Part 1

Case Study 3: Loan Amortization

Case Study 4: Creating a String Class

Case Study 5: High Adventure Travel Agency—Part 2

Case Study 6: High Adventure Travel Agency—Part 3

Case Study 7: Intersection of Sets

Case Study 8: Sales Commission

LOCATION OF VIDEONOTES IN THE TEXT

Chapter 1	Introduction to Flowcharting, p. 52 Designing a Program with Pseudocode, p. 52 Designing the Account Balance Program, p. 57 Predicting the Result of Problem 33, p. 58
Chapter 2	Using cout, p. 64 Variable Definitions, p. 70 Assignment Statements and Simple Math Expressions, p. 95 Solving the Restaurant Bill Problem, p. 113
Chapter 3	Reading Input with cin, p. 117 Formatting Numbers with setprecision, p. 145 Solving the Stadium Seating Problem, p. 176
Chapter 4	The if Statement, p. 188 The if/else statement, p. 200 The if/else if Statement, p. 210 Solving the Time Calculator Problem, p. 255
Chapter 5	The while Loop, p. 268 The for Loop, p. 283 Reading Data from a File, p. 310 Solving the Calories Burned Problem, p. 329
Chapter 6	Functions and Arguments, p. 339 Value-Returning Functions, p. 362 Solving the Markup Problem, p. 404
Chapter 7	Accessing Array Elements with a Loop, p. 418 Passing an Array to a Function, p. 445 Solving the Chips and Salsa Problem, p. 487
Chapter 8	The Binary Search, p. 498 The Selection Sort, p. 514 Solving the Charge Account Validation Modification Problem, p. 532
Chapter 9	Dynamically Allocating an Array, p. 563 Solving the Pointer Rewrite Problem, p. 586
Chapter 10	Writing a C-String-Handling Function, p. 617 More about the string Class, p. 623 Solving the Backward String Problem, p. 639

LOCATION OF VIDEONOTES IN THE TEXT *(continued)*



Chapter 11	Creating a Structure, p. 647 Passing a Structure to a Function, p. 663 Solving the Weather Statistics Problem, p. 691
Chapter 12	Passing File Stream Objects to Functions, p. 705 Working with Multiple Files, p. 718 Solving the File Encryption Filter Problem, p. 748
Chapter 13	Writing a Class, p. 758 Defining an Instance of a Class, p. 763 Solving the Employee Class Problem, p. 840
Chapter 14	Operator Overloading, p. 869 Class Aggregation, p. 898 Solving the NumDays Problem, p. 933
Chapter 15	Redefining a Base Class Function in a Derived Class, p. 968 Polymorphism, p. 979 Solving the Employee and ProductionWorker Classes Problem, p. 1013
Chapter 16	Throwing an Exception, p. 1022 Handling an Exception, p. 1022 Writing a Function Template, p. 1040 Solving the Exception Project Problem, p. 1060
Chapter 17	The array Container, p. 1064 Iterators, p. 1066 The vector Container, p. 1072 The map Container, p. 1086 The set Container, p. 1111 Function Objects and Lambda Expressions, p. 1139 The Course Information Problem, p. 1151
Chapter 18	Appending a Node to a Linked List, p. 1158 Inserting a Node in a Linked List, p. 1165 Deleting a Node from a Linked List, p. 1169 Solving the Member Insertion by Position Problem, p. 1194
Chapter 19	Storing Objects in an STL stack, p. 1225 Storing Objects in an STL queue, p. 1248 Solving the File Compare Problem, p. 1253
Chapter 20	Reducing a Problem with Recursion, p. 1260 Solving the Recursive Multiplication Problem, p. 1282
Chapter 21	Inserting a Node in a Binary Tree, p. 1294 Deleting a Node from a Binary Tree, p. 1300 Solving the Node Counter Problem, p. 1316

This page intentionally left blank

Preface

Welcome to *Starting Out with C++: From Control Structures through Objects, 9th edition*. 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 Ninth Edition

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

- The material on the Standard Template Library (STL) has been completely rewritten and expanded into its own chapter. Previously, Chapter 16 covered exceptions, templates, and gave brief coverage to the STL. In this edition, Chapter 16 covers exceptions and templates, and Chapter 17 is a new chapter dedicated to the STL. The new chapter covers the following topics:
 - The array and vector classes
 - The various types of iterators
 - Emplacement versus insertion
 - The `map`, `multimap`, and `unordered_map` Classes
 - The `set`, `multiset`, and `unordered_set` Classes
 - Sorting and searching algorithms
 - Permutation algorithms
 - Set algorithms
 - Using function pointers with STL algorithms
 - Function objects, or functors
 - Lambda expressions
- Chapter 2 now includes a discussion of alternative forms of variable initialization, including functional notation, and brace notation (also known as uniform initialization).

- Chapter 3 now mentions the `round` function, introduced in C++ 11.
- Chapter 7 now introduces array initialization much earlier.
- In Chapter 8, the bubble sort algorithm has been rewritten and improved.
- A new example of sorting and searching a vector of strings has been added to Chapter 8.
- In Chapter 9, the section on smart pointers now gives an overview of `shared_ptr`s and `weak_ptr`s, in addition to the existing coverage of `unique_ptr`s.
- In Chapter 10, a new *In the Spotlight* section on string tokenizing has been added.
- Chapter 10 now covers the string-to-number conversion functions that were introduced in C++ 11.
- The material on unions that previously appeared in Chapter 11 has been moved to Appendix K, available on the book's companion Website.
- Chapter 13 has new sections covering:
 - Member initialization lists.
 - In-place initialization.
 - Constructor delegation.
- Several new topics were added to Chapter 14, including:
 - Rvalue references and move semantics.
 - Checking for self-assignment when overloading the `=` operator.
 - Using member initialization lists in aggregate classes.
- Chapter 15 includes a new section on constructor inheritance.
- Several new programming problems have been added throughout the book.

Organization of the Text

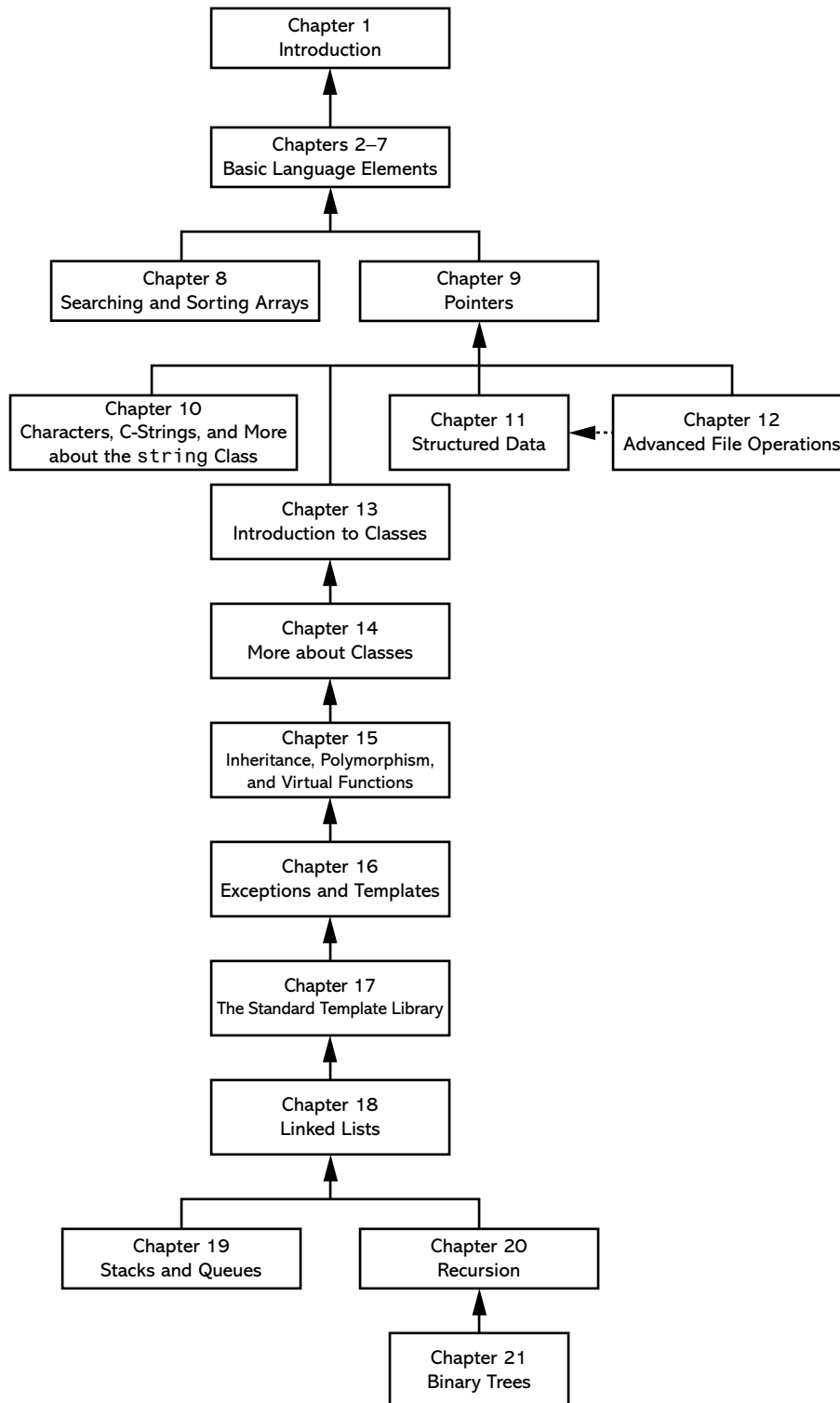
This text teaches C++ in a step-by-step fashion. Each chapter covers a major set of topics and builds knowledge as the student progresses through the book. Although the chapters can be easily taught in their existing sequence, some flexibility is provided. The diagram shown in Figure P-1 suggests possible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

Chapter 1 covers fundamental hardware, software, and programming concepts. You may choose to skip this chapter if the class is already familiar with 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 Chapter 9.

After Chapter 9 has been covered, Chapter 10, 11, 12, or 13 may be covered. (If you jump to Chapter 12 at this point, you will need to postpone Sections 12.8, 12.9, and 12.10 until Chapter 11 has been covered.) After Chapter 13, you may cover Chapters 14 through 18 in sequence. Next, you can proceed to either Chapter 19 or Chapter 20. Finally, Chapter 21 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.

Figure P-1 Chapter dependency chart

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.

Chapter 3: Expressions and Interactivity

In this chapter, the student learns to write programs that input and handle numeric, character, and 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 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: Loops and Files

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. Sequential file I/O is also introduced. The student learns to read and write text files, and use loops to process the data in a file.

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 and Vectors

In this chapter, the student learns to create and work with single and multi-dimensional 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: Searching and Sorting 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 More about the string Class

This chapter discusses various ways to process text at a detailed level. Library functions for testing and manipulating characters are introduced. C-strings are discussed, and the technique of storing C-strings in char arrays is covered. An extensive discussion of the `string` class methods is also given.

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 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, Polymorphism, and Virtual Functions

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 and Templates

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.

Chapter 17: The Standard Template Library

This chapter discusses the containers, iterators, and algorithms in the Standard Template Library (STL). The specific containers covered are the `array`, `vector`, `map`, `multimap`, `unordered_map`, `set`, `multiset`, and `unordered_set` classes. The student then learns about sorting, searching, permutation, and set algorithms. The chapter concludes with a discussion of function objects (functors) and lambda functions.

Chapter 18: 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 19: 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 20: 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 21: 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: The 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 available online at www.pearsonglobaleditions.com/gaddis.

Appendix C: Introduction to Flowcharting

A brief introduction to flowcharting. This tutorial discusses sequence, decision, 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: 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 command prompt.

Appendix G: Binary Numbers and Bitwise Operations

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

Appendix H: STL Algorithms

This appendix gives a summary of each of the function templates provided by the Standard Template Library (STL), and defined in the `<algorithm>` header file.

Appendix I: 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 J: Stream Member Functions for Formatting

Covers stream member functions for formatting such as `setf`

Appendix K: Unions

This appendix introduces unions. It describes the purpose of unions and the difference between a union and a `struct`, demonstrates how to declare a union and define a union variable, and shows example programs that use unions.

Appendix L: 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 M: 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 hundreds of 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.



In the Spotlight Each of these sections provides a programming problem and a detailed, step-by-step analysis showing the student how to solve it.



VideoNotes A series of online videos, developed specifically for this book, is available for viewing at www.pearsonglobaleditions.com/gaddis. Icons appear throughout the text alerting the student to videos about specific topics.



Checkpoints Checkpoints are questions placed throughout each chapter as a self-test study aid. Answers for all Checkpoint questions can be downloaded from the book's Website at www.pearsonglobaleditions.com/gaddis. This allows students to 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. 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 *Algorithm Workbench*, *Predict the Output*, and *Find the Errors* sections. Answers to the odd-numbered review questions and review exercises can be downloaded from the book's Website at www.pearsonglobaleditions.com/gaddis.

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

Available for download from the book's Website at www.pearsonglobaleditions.com/gaddis. This is an ongoing 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.

**C++11**

Throughout the text, new C++11 language features are introduced. Look for the C++11 icon to find these new features.

Supplements

Student Online Resources

Many student resources are available for this book from the publisher. The following items are available at www.pearsonglobaleditions.com/gaddis:

- The source code for each example program in the book
- Access to the book's VideoNotes
- A full set of appendices, including answers to the Checkpoint questions and answers to the odd-numbered review questions
- A collection of valuable Case Studies
- The complete Serendipity Booksellers Project

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 exercises 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.

Instructor Resources

The following supplements are available only to qualified instructors:

- 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 Pearson Instructor Resource Center (www.pearsonglobaleditions.com/gaddis) for information on how to access instructor resources.

Textbook Web site

Student and instructor resources, including links to download Microsoft® Visual Studio Express and other popular IDEs, for all the books in the Gaddis *Starting Out with* series can be accessed at the following URL:

<http://www.pearsonglobaleditions.com/gaddis>

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*
- *Starting Out with C++: Early Objects*
- *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</p> <p>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>INTRODUCTION OF DATA STRUCTURES AND RECURSION</p> <p>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</p> <p>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>INTRODUCTION OF DATA STRUCTURES AND RECURSION</p> <p>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 for the 9th Edition

Chia-Chin Chang
Lakeland College

William Duncan
Louisiana State University

Pranshu Gupta
DeSales University

Charles Hardnett
Gwinnett Technical College

Svetlana Marzelli
Atlantic Cape Community College

Jie Meichsner
St. Cloud State University

Ron Del Porto
Penn State Erie, The Behrend College

Lisa Rudnitsky
Baruch College

Reviewers for Previous Editions

Ahmad Abuhejleh
University of Wisconsin–River Falls

David Akins
El Camino College

Steve Allan
Utah State University

Vicki Allan
Utah State University

Karen M. Arlien
Bismark State College

Mary Astone
Troy University

Ijaz A. Awan
Savannah State University

Robert Baird
Salt Lake Community College

Don Biggerstaff
Fayetteville Technical Community College

Michael Bolton
Northeastern Oklahoma State University

Bill Brown
Pikes Peak Community College

Robert Burn
Diablo Valley 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

Michael Dixon
Sacramento City College

Jeanne Douglas
University of Vermont

Michael Dowell
Augusta State University

Qiang Duan
Penn State University—Abington

William E. Duncan
Louisiana State University

Daniel Edwards
Ohlone College

Judy Etchison
Southern Methodist University

Dennis Fairclough
Utah Valley State College

Xisheng Fang
Ohlone College

Mark Fienup
University of Northern Iowa

Richard Flint
North Central College

Ann Ford Tyson
Florida State University

Jeanette Gibbons
South Dakota State University

James Gifford
University of Wisconsin—Stevens Point

Leon Gleiberman
Touro College

Barbara Guillott
Louisiana State University

Ranette Halverson, Ph.D.
Midwestern State University

Ken Hang
Green River Community College

Carol Hannahs
University of Kentucky

Dennis Heckman
Portland Community College

Ric Heishman
George Mason University

Michael Hennessy
University of Oregon

Ilga Higbee
Black Hawk College

Patricia Hines
Brookdale Community College

Mike Holland
Northern Virginia Community College

Mary Hovik
Lehigh Carbon Community College

- Richard Hull
Lenoir-Rhyne College
- Kay Johnson
Community College of Rhode Island
- Chris Kardaras
North Central College
- Willard Keeling
Blue Ridge Community College
- A.J. Krygeris
Houston Community College
- Sheila Lancaster
Gadsden State Community College
- Ray Larson
Inver Hills Community College
- Michelle Levine
Broward College
- Jennifer Li
Ohlone College
- Norman H. Liebling
San Jacinto College
- Cindy Lindstrom
Lakeland College
- Zhu-qu Lu
University of Maine, Presque Isle
- Heidar Malki
University of Houston
- Debbie Mathews
J. Sargeant Reynolds Community College
- Rick Matzen
Northeastern State University
- Robert McDonald
East Stroudsburg University
- James McGuffee
Austin Community College
- Dean Mellas
Cerritos 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
- Farshad Ravanshad
Bergen Community College
- Susan Reeder
Seattle University
- Sandra Roberts
Snead College
- Lopa Roychoudhuri
Angelo State University
- Dolly Samson
Weber State University
- Ruth Sapir
SUNY Farmingdale
- Jason Schatz
City College of San Francisco
- Dr. Sung Shin
South Dakota State University
- Bari Siddique
University of Texas at Brownsville

William Slater
Collin County Community College

Shep Smithline
University of Minnesota

Richard Snyder
Lehigh Carbon Community College

Donald Southwell
Delta College

Caroline St. Claire
North Central College

Kirk Stephens
Southwestern Community College

Cherie Stevens
South Florida Community College

Dale Suggs
Campbell University

Mark Swanson
Red Wing Technical College

Ann Sudell Thorn
Del Mar College

Martha Tillman
College of San Mateo

Ralph Tomlinson
Iowa State University

David Topham
Ohlone College

Robert Tureman
Paul D. Camp Community College

Arisa K. Ude
Richland College

Peter van der Goes
Rose State College

Stewart Venit
California State University, Los Angeles

Judy Walters
North Central College

John H. Whipple
Northampton Community College

Aurelia Williams
Norfolk State University

Chadd Williams
Pacific University

Vida Winans
Illinois Institute of Technology

I would also like to thank my family and friends for their support in all of my projects. I am extremely fortunate to have Matt Goldstein as my editor, and Kristy Alaura as editorial assistant. Their guidance and encouragement made it a pleasure to write chapters and meet deadlines. I am also fortunate to have Demetrius Hall as my marketing manager. His hard work is truly inspiring, and he does a great job of getting this book out to the academic community. The production team, led by Sandra Rodriguez, worked tirelessly to make this book a reality. Thanks to you all!