# Brief Table of Contents

## Part One

### Fundamentals of C++ Programming

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Getting Started</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Data Types, Declarations, and Displays</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>Assignment and Interactive Input</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>Selection</td>
<td>137</td>
</tr>
<tr>
<td>5</td>
<td>Repetition</td>
<td>179</td>
</tr>
<tr>
<td>6</td>
<td>Modularity Using Functions</td>
<td>225</td>
</tr>
<tr>
<td>7</td>
<td>Arrays</td>
<td>291</td>
</tr>
<tr>
<td>8</td>
<td>Arrays and Pointers</td>
<td>341</td>
</tr>
<tr>
<td>9</td>
<td>I/O Streams and Data Files</td>
<td>387</td>
</tr>
</tbody>
</table>

## Part Two

### Object-Oriented Programming

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Introduction to Classes</td>
<td>433</td>
</tr>
<tr>
<td>11</td>
<td>Adding Functionality to Your Classes</td>
<td>491</td>
</tr>
</tbody>
</table>
**Brief Table of Contents**

<table>
<thead>
<tr>
<th>Chapter 12</th>
<th>539</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extending Your Classes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 13</th>
<th>573</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Standard Template Library</td>
<td></td>
</tr>
</tbody>
</table>

**Part Three**

**Additional Topics**

<table>
<thead>
<tr>
<th>Chapter 14</th>
<th>603</th>
</tr>
</thead>
<tbody>
<tr>
<td>The string Class and Exception Handling</td>
<td>605</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15</th>
<th>665</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strings as Character Arrays</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 16</th>
<th>701</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structures</td>
<td></td>
</tr>
</tbody>
</table>

**Appendixes**

<table>
<thead>
<tr>
<th>Appendix A</th>
<th>729</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator Precedence Table</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix B</th>
<th>731</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII Character Codes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit Operations</td>
<td></td>
</tr>
<tr>
<td><strong>Online Only</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix D</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating-Point Number Storage</td>
<td></td>
</tr>
<tr>
<td><strong>Online Only</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix E</th>
<th>733</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solutions to Selected Exercises</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>753</th>
</tr>
</thead>
</table>
Part One
Fundamentals of C++ Programming

Chapter 1
Getting Started

1.1 Introduction to Programming
   Algorithms and Procedures 3
   Classes and Objects 5
   Program Translation 9

1.2 Function and Class Names
   The main() Function 12

1.3 The cout Object 18

1.4 Programming Style
   Comments 22

1.5 Common Programming Errors 26

1.6 Chapter Summary 27

1.7 Chapter Supplement: Software Development
   Phase I: Development and Design 28
   Phase II: Documentation 32
   Phase III: Maintenance 33
   Backup 33

Chapter 2
Data Types, Declarations, and Displays

2.1 Data Types
   Integer Data Types 37
   Determining Storage Size 38
   Floating-Point Types 42
   Exponential Notation 45

2.2 Arithmetic Operations
   Expression Types 48
   Integer Division 51
   Negation 52
   Operator Precedence and Associativity 53

2.3 Variables and Declarations
   Declaration Statements 57
   Multiple Declarations 62
   Memory Allocation 64
Contents

2.4 Common Programming Errors 70
2.5 Chapter Summary 71
2.6 Chapter Supplement: Bits, Bytes, and Binary Number Representations 72
   Bits and Bytes 72
   Binary, Hexadecimal, and Octal Numbers 74

Chapter 3
Assignment and Interactive Input 79
3.1 Assignment Operators 79
   Coercion 84
   Assignment Variations 85
   Accumulating 86
   Counting 88
3.2 Formatted Output 93
   The setiosflags() Manipulator 97
   Hexadecimal and Octal I/O 99
3.3 Mathematical Library Functions 106
   Casts 111
3.4 Interactive Keyboard Input 117
   A First Look at User-Input Validation 121
3.5 Symbolic Constants 127
   Placement of Statements 128
3.6 Common Programming Errors 132
3.7 Chapter Summary 132
3.8 Chapter Supplement: Errors, Testing, and Debugging 133
   Compile-Time and Runtime Errors 134
   Syntax and Logic Errors 134
   Testing and Debugging 135

Chapter 4
Selection 137
4.1 Relational Expressions 137
   Logical Operators 139
   A Numerical Accuracy Problem 142
4.2 The if–else Statement 143
   Compound Statements 146
   Block Scope 148
   One-Way Selection 149
   Problems Associated with the if–else Statement 151
4.3 Nested if Statements 158
   The if–else Chain 159
## Contents

4.4 The `switch` Statement 167  
4.5 Common Programming Errors 173  
4.6 Chapter Summary 174  
4.7 Chapter Supplement: A Closer Look at Testing 176  

Chapter 5  
Repetition 179  
5.1 The `while` Statement 180  
5.2 Interactive `while` Loops 188  
Sentinels 194  
`break` and `continue` Statements 197  
The Null Statement 198  
5.3 The `for` Statement 201  
Interactive `for` Loops 208  
Nested Loops 209  
5.4 The `do-while` Statement 217  
Validity Checks 219  
5.5 Common Programming Errors 220  
5.6 Chapter Summary 222  

Chapter 6  
Modularity Using Functions 225  
6.1 Function and Parameter Declarations 226  
Function Prototypes 227  
Calling a Function 228  
Defining a Function 229  
Placement of Statements 234  
Function Stubs 234  
Functions with Empty Parameter Lists 235  
Default Arguments 236  
Reusing Function Names (Overloading) 237  
Function Templates 238  
6.2 Returning a Single Value 244  
Inline Functions 250  
Templates with a Return Value 251  
6.3 Returning Multiple Values 257  
Passing and Using Reference Parameters 258  
6.4 Variable Scope 267  
Scope Resolution Operator 271  
Misuse of Globals 272
Contents

6.5 Variable Storage Category 276
   Local Variable Storage Categories 277
   Global Variable Storage Categories 280
6.6 Common Programming Errors 285
6.7 Chapter Summary 285
6.8 Chapter Supplement: Generating Random Numbers 287
   Scaling 289

Chapter 7

Arrays 291

7.1 One-Dimensional Arrays 292
   Input and Output of Array Values 296
7.2 Array Initialization 303
7.3 Arrays as Arguments 307
7.4 Two-Dimensional Arrays 313
   Larger Dimensional Arrays 319
7.5 Common Programming Errors 323
7.6 Chapter Summary 324
7.7 Chapter Supplement: Searching and Sorting Methods 325
   Search Algorithms 325
   Sort Algorithms 333

Chapter 8

Arrays and Pointers 341

8.1 Introduction to Pointers 341
   Storing Addresses 344
   Using Addresses 345
   Declaring Pointers 346
   References and Pointers 348
8.2 Array Names as Pointers 354
   Dynamic Array Allocation 360
8.3 Pointer Arithmetic 364
   Pointer Initialization 368
8.4 Passing Addresses 369
   Passing Arrays 374
   Advanced Pointer Notation 378
8.5 Common Programming Errors 383
8.6 Chapter Summary 385
## Chapter 9

### I/O Streams and Data Files

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 I/O File Stream Objects and Methods</td>
<td>388</td>
</tr>
<tr>
<td>Files</td>
<td>388</td>
</tr>
<tr>
<td>File Stream Objects</td>
<td>389</td>
</tr>
<tr>
<td>File Stream Methods</td>
<td>390</td>
</tr>
<tr>
<td>9.2 Reading and Writing Text Files</td>
<td>403</td>
</tr>
<tr>
<td>Reading from a Text File</td>
<td>406</td>
</tr>
<tr>
<td>Standard Device Files</td>
<td>411</td>
</tr>
<tr>
<td>Other Devices</td>
<td>412</td>
</tr>
<tr>
<td>9.3 Random File Access</td>
<td>416</td>
</tr>
<tr>
<td>9.4 File Streams as Function Arguments</td>
<td>419</td>
</tr>
<tr>
<td>9.5 Common Programming Errors</td>
<td>423</td>
</tr>
<tr>
<td>9.6 Chapter Summary</td>
<td>423</td>
</tr>
<tr>
<td>9.7 Chapter Supplement: The <code>iostream</code> Class Library</td>
<td>426</td>
</tr>
<tr>
<td>File Stream Transfer Mechanism</td>
<td>426</td>
</tr>
<tr>
<td>Components of the <code>iostream</code> Class Library</td>
<td>426</td>
</tr>
<tr>
<td>In-Memory Formatting</td>
<td>428</td>
</tr>
</tbody>
</table>

## Part Two

### Object-Oriented Programming

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 10</td>
<td>433</td>
</tr>
<tr>
<td>Introduction to Classes</td>
<td>433</td>
</tr>
<tr>
<td>10.1 Object-Based Programming</td>
<td>433</td>
</tr>
<tr>
<td>A Class Is a Plan</td>
<td>436</td>
</tr>
<tr>
<td>From Recipe to Class</td>
<td>436</td>
</tr>
<tr>
<td>10.2 Creating Your Own Classes</td>
<td>439</td>
</tr>
<tr>
<td>Class Construction</td>
<td>440</td>
</tr>
<tr>
<td>Terminology</td>
<td>448</td>
</tr>
<tr>
<td>10.3 Constructors</td>
<td>452</td>
</tr>
<tr>
<td>Calling Constructors</td>
<td>454</td>
</tr>
<tr>
<td>Overloaded and Inline Constructors</td>
<td>455</td>
</tr>
<tr>
<td>Destructors</td>
<td>458</td>
</tr>
<tr>
<td>Arrays of Objects</td>
<td>459</td>
</tr>
<tr>
<td>10.4 Examples</td>
<td>463</td>
</tr>
<tr>
<td>Example 1: Constructing a Room Object</td>
<td>463</td>
</tr>
<tr>
<td>Example 2: Constructing an Elevator Object</td>
<td>467</td>
</tr>
</tbody>
</table>
Contents

10.5 Class Scope and Duration Categories 473
  Static Class Members 474
  Friend Functions 478

10.6 Common Programming Errors 482

10.7 Chapter Summary 482

10.8 Chapter Supplement: Thinking in Terms of Objects 484
  Representing Problems with Models 485
  Modeling Classes 486

Chapter 11
Adding Functionality to Your Classes 491

11.1 Creating Class Operators 491
  Assignment Operator 499
  Copy Constructors 500
  Base/Member Initialization 503
  Operator Functions as Friends 504

11.2 How Methods Are Shared 508
  The this Pointer 509
  The Assignment Operator Revisited 511
  Objects as Arguments 514
  Notation 515

11.3 Data Type Conversions 519
  Built-in to Built-in Conversion 520
  Class to Built-in Conversion 520
  Built-in to Class Conversion 522
  Class to Class Conversion 525

11.4 Two Useful Alternatives: operator() and operator[] 529

11.5 Common Programming Errors 533

11.6 Chapter Summary 533

11.7 Chapter Supplement: Insides and Outsides 535
  Abstraction and Encapsulation 537
  Code Extensibility 537

Chapter 12
Extending Your Classes 539

12.1 Class Inheritance 540
  Access Specifications 541

12.2 Polymorphism 547

12.3 Dynamic Object Creation and Deletion 552

12.4 Pointers as Class Members 562
  Assignment Operators and Copy Constructors Reconsidered 565
### Contents

12.5 Common Programming Errors 568  
12.6 Chapter Summary 568  
12.7 Chapter Supplement: UML Class and Object Diagrams 569  
  Class and Object Diagrams 570  

**Chapter 13**  
The Standard Template Library 573  

#### 13.1 The Standard Template Library 574  

#### 13.2 Linked Lists 579  
  Using the STL `list` Class 581  
  Using User-Defined Objects 586  

#### 13.3 Stacks 590  
  Stack Implementation with the `deque` Class 592  

#### 13.4 Queues 598  
  Queue Implementation with the `deque` Class 599  

#### 13.5 Common Programming Errors 602  

#### 13.6 Chapter Summary 602  

**Part Three**  
Additional Topics 603  

**Chapter 14**  
The `string` Class and Exception Handling 605  

#### 14.1 The `string` Class 606  
  `string` Class Functions 607  
  String Input and Output 609  
  String Processing 613  

#### 14.2 Character Manipulation Methods 622  
  Character I/O 627  
  A Second Look at User-Input Validation 632  

#### 14.3 Exception Handling 634  

#### 14.4 Exceptions and File Checking 640  
  Opening Multiple Files 644  

#### 14.5 Input Data Validation 649  

#### 14.6 Common Programming Errors 656  

#### 14.7 Chapter Summary 656  

#### 14.8 Chapter Supplement: Namespaces and Creating a Personal Library 657
### Chapter 15

**Strings as Character Arrays** 665

15.1 C-String Fundamentals 666  
  C-String Input and Output 666  
  C-String Processing 670  

15.2 Pointers and C-String Library Functions 675  
  Library Functions 681  
  Character-Handling Functions 684  
  Conversion Functions 688  

15.3 C-String Definitions and Pointer Arrays 691  
  Pointer Arrays 693  

15.4 Common Programming Errors 698  

15.5 Chapter Summary 698  

### Chapter 16

**Data Structures** 701

16.1 Single Structures 702  
16.2 Arrays of Structures 708  
16.3 Structures as Function Arguments 712  
  Passing a Pointer 716  
  Returning Structures 718  

16.4 Dynamic Structure Allocation 721  

16.5 Unions 724  

16.6 Common Programming Errors 727  

16.7 Chapter Summary 728  

### Appendixes

**Appendix A**

Operator Precedence Table 729  

**Appendix B**

ASCII Character Codes 731  

**Appendix C**

Bit Operations  
  *Online Only*  

**Appendix D**

Floating-Point Number Storage  
  *Online Only*  

**Appendix E**

Solutions to Selected Exercises 733  

Index 753
The main goal of this fourth edition of *A First Book of C++* remains the same as in previous editions: to introduce, develop, and reinforce well-organized programming skills using C++. All topics are presented in a clear, unambiguous, and accessible manner to beginning students. Students should be familiar with fundamental algebra, but no other prerequisites are assumed.

Therefore, like the first three editions, this new edition begins by providing a strong foundation in structured programming. This foundation is then expanded to an object-oriented design and programming approach in a pedagogically sound, achievable progression. In addition to a number of minor changes throughout the book, the major changes in this edition are the following:

- Part I has been restructured to include arrays, files, and pointers, so it can be used as the basis for a complete introductory semester course in C++.
- The four chapters covering object-oriented programming have been revised and moved to Part II so that they form a logical continuation from structured programming to object-oriented programming.
- More than 50 new exercises have been added, and all exercises are labeled to indicate their function (Review, Practice, Program, Modify, Debug, Desk check, or For thought).
- Three new Chapter Supplements have been added to introduce the fundamentals of object-oriented design and the Unified Modeling Language (UML).
- A complete set of case studies has been added and is available on the Cengage Web site, login.cengage.com, for instructors to distribute.

The following features from the third edition have been retained:

- Fundamentals of software engineering are discussed from both procedural and object-oriented viewpoints.
- Each chapter contains a Common Programming Errors section that describes problems beginning C++ programmers typically encounter.
- The ANSI/ISO C++ *iostream* library and *namespace* mechanism are used in all programs.
- Exception handling is discussed in a separate section, with practical applications of exception handling included throughout the book.
- The C++ *string* class is covered.
- A thorough discussion is included of input data validation and functions to check the numerical data type of input items and allow reentering invalid numerical types.

In practical terms, this book has been written to support both a one- and two-semester technical C++ programming course; the only prerequisite is that students should be familiar with fundamental algebra. This book is constructed to be flexible enough so that instructors can mold the book to their preferences for topic sequence. This flexibility is achieved in the following ways.

Part I includes the basic structured syntax, flow control, and modularity topics needed for a thorough understanding of C++’s structural features. With the topics of arrays (Chapter 7) and files (Chapter 9) moved to Part I, this part now provides a comprehensive one-semester
course. As Chapters 7 and 9 have been written to depend only on Chapters 1 through 6, their order of presentation (arrays first and files second, or vice versa) is entirely up to the instructor’s discretion. With time permitting, the basics of classes, introduced in Chapter 10, can also be covered to create a one-semester course with an introduction to object-oriented programming. Figure 1 illustrates this one-semester topic dependency, and Figure 2 shows the topic dependency chart for the entire book.

![Figure 1](image1.png) Topic dependency for a one-semester course

![Figure 2](image2.png) Topic dependency chart

**Distinctive Features of This Book**

**Writing Style** One thing I have found to be essential in my own classes is that after the instructor sets the stage in class, the assigned book must continue to encourage, nurture, and assist students in acquiring and “owning” the material. To do this, the book must be written in a manner that makes sense to students. My primary concern, and one of the distinctive features of this book, is that it has been written for students. Therefore, I believe the writing style used to convey the concepts is one of the most important aspects of this book.
Software Engineering Rather than simply introduce students to programming in C++, this book introduces students to the fundamentals of software engineering, from both a proce- dural and object-oriented viewpoint. It begins with a discussion of these two programming approaches in Section 1.1 and is reinforced throughout the book.

Introduction to References and Pointers A unique feature of my book *A First Book of ANSI C* was introducing pointer concepts early by displaying addresses of variables and then using other variables to store these addresses. This approach always seemed a more logical method of understanding pointers than the indirection description in vogue at the time *A First Book of ANSI C* was released.

I have since been pleased to see that using an output function to display addresses has become a standard way of introducing pointers. Although this approach is no longer a unique feature of this book, I’m proud of its presentation and continue to use it in this book. References are also introduced early, in Chapter 6, before the introduction of pointers in Chapter 8.

Program Testing Every C++ program in this book has been compiled and run successfully and has been quality-assurance tested with Microsoft Visual C++ 2010. Source code for all programs is available for student download at [www.cengagebrain.com](http://www.cengagebrain.com). Using this source code enables students to experiment with and extend the existing programs and modify them more easily, as required for a number of end-of-section exercises.

Pedagogical Features
To facilitate the goal of making C++ accessible as a first-level course, the following pedagogical features have been incorporated into the book.

Point of Information Boxes These shaded boxes in each chapter highlight important con- cepts, useful technical points, programming tips, and tricks used by professional programmers.

End-of-Section Exercises Almost every section in the book contains numerous and diverse skill-building and programming exercises. In addition, solutions to selected exercises are given in Appendix E.

Pseudocode Descriptions Pseudocode is used throughout the book. Flowchart symbols are introduced but are used only in illustrating flow-of-control constructs.

Common Programming Errors and Chapter Summary Each chapter ends with a section on common programming errors and a summary of the main topics covered in the chapter.

Appendixes This book includes appendixes on operator precedence, ASCII codes, and solutions to selected exercises. Additional appendixes on bit operations and floating-point number storage are available for student download at [www.cengagebrain.com](http://www.cengagebrain.com).
Note to students: Microsoft offers a free C++ compiler and development system called Microsoft Visual C++ Express 2010. To get this development system, go to www.microsoft.com/express/Downloads/#2010-Visual-CPP and select English as the language. The vc_web file is downloaded automatically to your Downloads folder. (If you don’t have this folder, do a search to see where the file was downloaded.) After this file is downloaded, double-click it to install Visual C++ Express 2010.

All programs in this book can be run as Visual C++ Express 2010 CLR Console Applications or Win32 Console Applications programs, with two additions:

- The code line #include "stdafx.h" must be added at the beginning of the program.
- The code line cin.ignore(); must be included before the return statement.

These added code lines hold the window open after the program runs so that you can view it. Pressing Enter terminates the program and closes the window. For example, to compile and run Program 1.1 in this book, you should enter the program in Visual C++ Express 2010 as follows:

```c++
#include "stdafx.h"    // needed for Visual C++ Express 2010
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello there world!";

    cin.ignore(); // needed for Visual C++ Express 2010

    return 0;
}
```

All the solution files provided for this book (and available to instructors) include these two extra code lines. In programs requiring user input, a second cin.ignore() statement is included to prevent the Enter key used when entering data from closing the window.

Supplemental Materials

The following supplemental materials are available to instructors when this book is used in a classroom setting. Most of the materials are also available on the Instructor Resources CD.

Electronic Instructor’s Manual. The Instructor’s Manual that accompanies this book includes the following:

- Additional instructional material to assist in class preparation, including suggestions for lecture topics
- Solutions to all end-of-section exercises

ExamView. This book is accompanied by ExamView, a powerful testing software package that allows instructors to create and administer printed, computer (LAN-based), and Internet exams. ExamView includes hundreds of questions that correspond to the topics covered in this
book, enabling students to generate detailed study guides that include page references for further review. These computer-based and Internet testing components allow students to take exams at their computers and save instructors time because each exam is graded automatically. The Test Bank is also available in WebCT and Blackboard formats.

**PowerPoint Presentations.** This book comes with Microsoft PowerPoint slides for each chapter. They are included as a teaching aid for classroom presentations, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics they introduce to the class.

**Source Code.** The source code for this book is available for students at [www.cengagebrain.com](http://www.cengagebrain.com) and is also available on the Instructor Resources CD.

**Solution Files.** The solution files for all programming exercises are available at [login.cengage.com](http://login.cengage.com) and on the Instructor Resources CD.

**Case Studies.** A complete set of case studies, keyed to Chapters 1 through 10, are available to instructors at [login.cengage.com](http://login.cengage.com).
Acknowledgments
The writing of this fourth edition is a direct result of the success (and limitations) of the previous editions. In this regard, my most heartfelt acknowledgment and appreciation is to the instructors and students who found the previous editions to be of service in their quests to teach and learn C++.

Next, I would like to thank Alyssa Pratt, my Senior Product Manager at Course Technology. In addition to her continuous faith and encouragement, her ideas and partnership were instrumental in creating this book. After the writing process was completed, the task of turning the final manuscript into a book depended on many people other than myself. For this, I especially want to thank my developmental editor, Lisa Lord, who provided an outstanding job. Her editing so dovetailed with both the spirit and idiosyncrasies of my own writing style that it was an absolute pleasure working with her. She stayed true to what I was attempting to achieve while patiently going through both the technical and grammatical content. A truly incredible feat! This editing was supplemented by the equally detailed work of my colleague Professor Joan Zucker Hoffman. Finally, I would like to thank Serge Palladino from Course Technology’s MQA Department, who was the validation tester for this book, as well as GEX Publishing Services, especially the interior designer. The dedication of this team of people was extremely important to me, and I am very grateful to them.

The following reviewers provided extensive, extremely useful, and detailed information and corrections that made this edition better and more accurate. No matter how careful I was, each reviewer pointed out something that I missed or could be stated better. I am very thankful to them. Naturally, all errors rest squarely on my shoulders, but these reviewers made the load much easier: Lynne Band, Middlesex Community College, and Alexandra Vaschillo, Lake Washington Technical College.

I would also like to acknowledge, with extreme gratitude, the wonderful academic environment for learning and teaching created at Fairleigh Dickinson University—starting with the President, Dr. Michael Adams, followed through in the academic departments by the university and campus provosts, Dr. Joseph Kiernan and Dr. Kenneth Greene, and finally to the encouragement and support provided by my dean, Dr. William Moore, and my chairperson, Dr. Paul Yoon. Without their support, this book could not have been written.

Finally, I deeply appreciate the patience, understanding, and love provided by my friend, wife, and partner, Rochelle.

Gary Bronson