

Fifth
Edition

Starting Out with



Programming Logic & Design

Fifth
Edition

Starting Out with

Programming Logic & Design

Tony Gaddis
Haywood Community College



330 Hudson Street, NY 10013

Senior Vice President Courseware Portfolio Management: Marcia Horton
Director, Portfolio Management: Engineering, Computer Science & Global Editions: Julian Partridge
Specialist, Higher Ed Portfolio Management: Matt Goldstein
Portfolio Management Assistant: Meghan Jacoby
Managing Content Producer: Scott Disanno
Content Producer: Carole Snyder
Web Developer: Steve Wright
Rights and Permissions Manager: Ben Ferrini
Manufacturing Buyer, Higher Ed, Lake Side Communications Inc (LSC): Maura Zaldivar-Garcia
Inventory Manager: Ann Lam
Product Marketing Manager: Yvonne Vannatta
Field Marketing Manager: Demetrius Hall
Marketing Assistant: Jon Bryant
Cover Designer: Joyce Wells, jWellsDesign
Full-Service Project Management: Mohamed Hameed, SPi Global
Composition: SPi Global

Copyright © 2019 by Pearson Education, Inc. or its affiliates. All Rights Reserved. Printed in the United States of America. This publication is protected by copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit www.pearsoned.com/permissions/.

Unless otherwise indicated herein, any third-party trademarks that may appear in this work are the property of their respective owners and any references to third-party trademarks, logos or other trade dress are for demonstrative or descriptive purposes only. Such references are not intended to imply any sponsorship, endorsement, authorization, or promotion of Pearson's products by the owners of such marks, or any relationship between the owner and Pearson Education, Inc. or its affiliates, authors, licensees or distributors.

Many of the designations by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided "as is" without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability. Whether express, implied or statutory, fitness for a particular.

Purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract. Negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time partial screen shots may be viewed in full within the software version specified.

Microsoft® Windows®, and Microsoft Office® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

Library of Congress Cataloging-in-Publication Data

Names: Gaddis, Tony, author.

Title: Starting out with programming logic and design / with Tony Gaddis, Haywood Community College.

Other titles: Starting out with programming logic & design

Description: Fifth edition. | New York, NY Pearson Education, Inc., [2019] | Earlier editions published under title: Starting out with programming logic & design. | Includes index.

Identifiers: LCCN 2017054391 | ISBN 978-0-13-480115-5 | ISBN 0-13-480115-6

Subjects: LCSH: Computer programming.

Classification: LCC QA76.6 .G315 2019 | DDC 005.1—dc23 LC record available at <https://lcn.loc.gov/2017054391>

1 18



ISBN 10: 0-13-480115-6

ISBN 13: 978-0-13-480115-5

Brief Contents

	Preface	xiii
	Acknowledgments	xxi
	About the Author	xxiii
Chapter 1	Introduction to Computers and Programming	1
Chapter 2	Input, Processing, and Output	27
Chapter 3	Modules	103
Chapter 4	Decision Structures and Boolean Logic	157
Chapter 5	Repetition Structures	217
Chapter 6	Functions	283
Chapter 7	Input Validation	333
Chapter 8	Arrays	351
Chapter 9	Sorting and Searching Arrays	419
Chapter 10	Files	469
Chapter 11	Menu-Driven Programs	543
Chapter 12	Text Processing	595
Chapter 13	Recursion	623
Chapter 14	Object-Oriented Programming	649
Chapter 15	GUI Applications and Event-Driven Programming	715
Appendix A	ASCII/Unicode Characters	747
Appendix B	Flowchart Symbols	749
Appendix C	Pseudocode Reference	751
Appendix D	Converting Decimal Numbers to Binary	765
Appendix E	Answers to Checkpoint Questions	767
	Index	783

Contents

Preface xiii

Acknowledgments xxi

About the Author xxiii

Chapter 1 Introduction to Computers and Programming 1

1.1	Introduction.....	1
1.2	Hardware.....	2
1.3	How Computers Store Data.....	7
1.4	How a Program Works.....	12
1.5	Types of Software.....	20
	Review Questions.....	21

Chapter 2 Input, Processing, and Output 27

2.1	Designing a Program.....	27
2.2	Output, Input, and Variables.....	34
2.3	Variable Assignment and Calculations.....	43
	IN THE SPOTLIGHT: Calculating Cell Phone Overage Fees.....	47
	IN THE SPOTLIGHT: Calculating a Percentage.....	49
	IN THE SPOTLIGHT: Calculating an Average.....	52
	IN THE SPOTLIGHT: Converting a Math Formula to a Programming Statement... ..	55
2.4	Variable Declarations and Data Types.....	57
2.5	Named Constants.....	62
2.6	Hand Tracing a Program.....	64
2.7	Documenting a Program.....	65
	IN THE SPOTLIGHT: Using Named Constants, Style Conventions, and Comments.....	66
2.8	Designing Your First Program.....	68
2.9	Focus on Languages: Java, Python, and C++.....	72
	Review Questions.....	92
	Debugging Exercises.....	97
	Programming Exercises.....	98

Chapter 3 Modules 103

3.1 Introduction to Modules	103
3.2 Defining and Calling a Module	106
IN THE SPOTLIGHT: Defining and Calling Modules	112
3.3 Local Variables	117
3.4 Passing Arguments to Modules	120
IN THE SPOTLIGHT: Passing an Argument to a Module	124
IN THE SPOTLIGHT: Passing an Argument by Reference	130
3.5 Global Variables and Global Constants	133
IN THE SPOTLIGHT: Using Global Constants	134
3.6 Focus on Languages: Java, Python, and C++	138
Review Questions	149
Debugging Exercises	153
Programming Exercises	154

Chapter 4 Decision Structures and Boolean Logic 157

4.1 Introduction to Decision Structures	157
IN THE SPOTLIGHT: Using the If-Then Statement	164
4.2 Dual Alternative Decision Structures	167
IN THE SPOTLIGHT: Using the If-Then-Else Statement	168
4.3 Comparing Strings	173
4.4 Nested Decision Structures	177
IN THE SPOTLIGHT: Multiple Nested Decision Structures	180
4.5 The Case Structure	184
IN THE SPOTLIGHT: Using a Case Structure	187
4.6 Logical Operators	189
4.7 Boolean Variables	196
4.8 Focus on Languages: Java, Python, and C++	197
Review Questions	209
Debugging Exercises	213
Programming Exercises	213

Chapter 5 Repetition Structures 217

5.1 Introduction to Repetition Structures	217
5.2 Condition-Controlled Loops: While, Do-While, and Do-Until	218
IN THE SPOTLIGHT: Designing a While Loop	223
IN THE SPOTLIGHT: Designing a Do-While Loop	232
5.3 Count-Controlled Loops and the For Statement	237
IN THE SPOTLIGHT: Designing a Count-Controlled Loop with the For Statement	245
5.4 Calculating a Running Total	255

5.5	Sentinels	259
	IN THE SPOTLIGHT: Using a Sentinel	260
5.6	Nested Loops	263
5.7	Focus on Languages: Java, Python, and C++	266
	Review Questions	275
	Debugging Exercises	278
	Programming Exercises	279

Chapter 6 **Functions 283**

6.1	Introduction to Functions: Generating Random Numbers	283
	IN THE SPOTLIGHT: Using Random Numbers	287
	IN THE SPOTLIGHT: Using Random Numbers to Represent Other Values	290
6.2	Writing Your Own Functions	292
	IN THE SPOTLIGHT: Modularizing with Functions	298
6.3	More Library Functions	307
6.4	Focus on Languages: Java, Python, and C++	317
	Review Questions	324
	Debugging Exercises	327
	Programming Exercises	328

Chapter 7 **Input Validation 333**

7.1	Garbage In, Garbage Out	333
7.2	The Input Validation Loop	334
	IN THE SPOTLIGHT: Designing an Input Validation Loop	336
7.3	Defensive Programming	341
7.4	Focus on Languages: Java, Python, and C++	342
	Review Questions	346
	Debugging Exercises	348
	Programming Exercises	349

Chapter 8 **Arrays 351**

8.1	Array Basics	351
	IN THE SPOTLIGHT: Using Array Elements in a Math Expression	358
8.2	Sequentially Searching an Array	365
8.3	Processing the Contents of an Array	371
	IN THE SPOTLIGHT: Processing an Array	378
8.4	Parallel Arrays	385
	IN THE SPOTLIGHT: Using Parallel Arrays	386
8.5	Two-Dimensional Arrays	390
	IN THE SPOTLIGHT: Using a Two-Dimensional Array	393
8.6	Arrays of Three or More Dimensions	398

8.7 Focus on Languages: Java, Python, and C++	399
Review Questions	409
Debugging Exercises	412
Programming Exercises	413

Chapter 9 **Sorting and Searching Arrays 419**

9.1 The Bubble Sort Algorithm	419
IN THE SPOTLIGHT: Using the Bubble Sort Algorithm	426
9.2 The Selection Sort Algorithm	433
9.3 The Insertion Sort Algorithm	439
9.4 The Binary Search Algorithm	445
IN THE SPOTLIGHT: Using the Binary Search Algorithm	449
9.5 Focus on Languages: Java, Python, and C++	451
Review Questions	464
Debugging Exercises	467
Programming Exercises	467

Chapter 10 **Files 469**

10.1 Introduction to File Input and Output	469
10.2 Using Loops to Process Files	481
IN THE SPOTLIGHT: Working with Files	486
10.3 Using Files and Arrays	490
10.4 Processing Records	491
IN THE SPOTLIGHT: Adding and Displaying Records	496
IN THE SPOTLIGHT: Searching for a Record	500
IN THE SPOTLIGHT: Modifying Records	502
IN THE SPOTLIGHT: Deleting Records	506
10.5 Control Break Logic	509
IN THE SPOTLIGHT: Using Control Break Logic	510
10.6 Focus on Languages: Java, Python, and C++	516
Review Questions	536
Debugging Exercises	539
Programming Exercises	540

Chapter 11 **Menu-Driven Programs 543**

11.1 Introduction to Menu-Driven Programs	543
11.2 Modularizing a Menu-Driven Program	554
11.3 Using a Loop to Repeat the Menu	559
IN THE SPOTLIGHT: Designing a Menu-Driven Program	564
11.4 Multiple-Level Menus	578
11.5 Focus on Languages: Java, Python, and C++	584

Review Questions	590
Programming Exercises	592

Chapter 12 Text Processing 595

12.1 Introduction	595
12.2 Character-by-Character Text Processing	597
IN THE SPOTLIGHT: Validating a Password	600
IN THE SPOTLIGHT: Formatting and Unformatting Telephone Numbers	606
12.3 Focus on Languages: Java, Python, and C++	611
Review Questions	617
Debugging Exercises	619
Programming Exercises	620

Chapter 13 Recursion 623

13.1 Introduction to Recursion	623
13.2 Problem Solving with Recursion	626
13.3 Examples of Recursive Algorithms	630
13.4 Focus on Languages: Java, Python, and C++	640
Review Questions	645
Programming Exercises	647

Chapter 14 Object-Oriented Programming 649

14.1 Procedural and Object-Oriented Programming	649
14.2 Classes	653
14.3 Using the Unified Modeling Language to Design Classes	664
14.4 Finding the Classes and Their Responsibilities in a Problem	667
IN THE SPOTLIGHT: Finding the Classes in a Problem	667
IN THE SPOTLIGHT: Determining Class Responsibilities	671
14.5 Inheritance	677
14.6 Polymorphism	685
14.7 Focus on Languages: Java, Python, and C++	689
Review Questions	707
Programming Exercises	710

Chapter 15 GUI Applications and Event-Driven Programming 715

15.1 Graphical User Interfaces	715
15.2 Designing the User Interface for a GUI Program	718
IN THE SPOTLIGHT: Designing a Window	723
15.3 Writing Event Handlers	725

IN THE SPOTLIGHT: Designing an Event Handler	728
15.4 Designing Apps for Mobile Devices	731
15.5 Focus on Languages: Java, Python, and C++	740
Review Questions	741
Programming Exercises	744

Appendix A	ASCII/Unicode Characters	747
-------------------	---------------------------------	------------

Appendix B	Flowchart Symbols	749
-------------------	--------------------------	------------

Appendix C	Pseudocode Reference	751
-------------------	-----------------------------	------------

Appendix D	Converting Decimal Numbers to Binary	765
-------------------	---	------------

Appendix E	Answers to Checkpoint Questions	767
-------------------	--	------------

Index	783
--------------	------------

Preface

Welcome to *Starting Out with Programming Logic and Design*, Fifth Edition. This book uses a language-independent approach to teach programming concepts and problem-solving skills, without assuming any previous programming experience. By using easy-to-understand pseudocode, flowcharts, and other tools, the student learns how to design the logic of programs without the complication of language syntax.

Fundamental topics such as data types, variables, input, output, control structures, modules, functions, arrays, and files are covered as well as object-oriented concepts, GUI development, and event-driven programming. As with all the books in the *Starting Out With . . .* series, this text is written in clear, easy-to-understand language that students find friendly and inviting.

Each chapter presents a multitude of program design examples. Short examples that highlight specific programming topics are provided, as well as more involved examples that focus on problem solving. Each chapter includes at least one *In the Spotlight section* that provides step-by-step analysis of a specific problem and demonstrates a solution to that problem.

This book is ideal for a programming logic course that is taught as a precursor to a language-specific introductory programming course, or for the first part of an introductory programming course in which a specific language is taught.

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:

- A new section titled *Focus on Languages: Java, Python, and C++* appears at the end of Chapters 2–15. This section discusses how the chapter's main topics are implemented in the Java, Python, and C++ programming languages. This section gives code snippet examples and complete programs in each of these languages. It is a valuable tool that students can use to learn the concepts of each chapter in one or more of these programming languages.
- A new section on the `Init()` module has been added to Chapter 15. This module is a startup module in a GUI application, similar to the `start` method in a JavaFX application, or the `Form_Load` handler in a Windows Forms application.
- A new section titled *Designing Apps for Mobile Devices* has been added to Chapter 15. This new section discusses some of the common issues developers face when designing apps for mobile devices. The student is also introduced to coding concepts related to a mobile device's special hardware capabilities, such as sending and receiving text messages, making and receiving phone calls, and detecting the device's location. Several pseudocode examples are given for a simulated smartphone.

- New motivational programming exercises have been added to several chapters.
- The book's Language Reference Guides have been updated. All of the book's Language Reference Guides are available on the book's resource site at: www.pearson.com/cs-resources.

Brief Overview of Each Chapter

Chapter 1: Introduction to Computers and Programming

This chapter begins by giving a concise and easy-to-understand explanation of how computers work, how data is stored and manipulated, and why we write programs in high-level languages.

Chapter 2: Input, Processing, and Output

This chapter introduces the program development cycle, data types, variables, and sequence structures. The student learns to use pseudocode and flowcharts to design simple programs that read input, perform mathematical operations, and produce screen output.

Chapter 3: Modules

This chapter demonstrates the benefits of modularizing programs and using the top-down design approach. The student learns to define and call modules, pass arguments to modules, and use local variables. Hierarchy charts are introduced as a design tool.

Chapter 4: Decision Structures and Boolean Logic

In this chapter students explore relational operators and Boolean expressions and are shown how to control the flow of a program with decision structures. The If-Then, If-Then-Else, and If-Then-Else If statements are covered. Nested decision structures, logical operators, and the case structure are also discussed.

Chapter 5: Repetition Structures

This chapter shows the student how to use loops to create repetition structures. The While, Do-While, Do-Until, and For loops are presented. Counters, accumulators, running totals, and sentinels are also discussed.

Chapter 6: Functions

This chapter begins by discussing common library functions, such as those for generating random numbers. After learning how to call library functions and how to use values returned by functions, the student learns how to define and call his or her own functions.

Chapter 7: Input Validation

This chapter discusses the importance of validating user input. The student learns to write input validation loops that serve as error traps. Defensive programming and the importance of anticipating obvious as well as unobvious errors is discussed.

Chapter 8: Arrays

In this chapter the student learns to create and work with one- and two-dimensional arrays. Many examples of array processing are provided including examples illustrating how to find the sum, average, and 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.

Chapter 9: Sorting and Searching Arrays

In this chapter the student learns the basics of sorting arrays and searching for data stored in them. The chapter covers the bubble sort, selection sort, insertion sort, and binary search algorithms.

Chapter 10: Files

This chapter introduces sequential file input and output. The student learns to read and write large sets of data, store data as fields and records, and design programs that work with both files and arrays. The chapter concludes by discussing control break processing.

Chapter 11: Menu-Driven Programs

In this chapter the student learns to design programs that display menus and execute tasks according to the user's menu selection. The importance of modularizing a menu-driven program is also discussed.

Chapter 12: Text Processing

This chapter discusses text processing at a detailed level. Algorithms that step through the individual characters in a string are discussed, and several common library functions for character and text processing are introduced.

Chapter 13: Recursion

This chapter discusses recursion and its use in problem solving. A visual trace of recursive calls is provided, and recursive applications are discussed. Recursive algorithms for many tasks are presented, such as finding factorials, finding a greatest common denominator (GCD), summing a range of values in an array, and performing a binary search. The classic Towers of Hanoi example is also presented.

Chapter 14: Object-Oriented Programming

This chapter compares procedural and object-oriented programming practices. It covers the fundamental concepts of classes and objects. Fields, methods, access specification, constructors, accessors, and mutators are discussed. The student learns how to model classes with UML and how to find the classes in a particular problem.

Chapter 15: GUI Applications and Event-Driven Programming

This chapter discusses the basic aspects of designing a GUI application. Building graphical user interfaces with visual design tools (such as Visual Studio® or NetBeans™) is discussed. The student learns how events work in a GUI application and how to write event handlers.

Appendix A: ASCII/Unicode Characters

This appendix lists the ASCII character set, which is the same as the first 127 Unicode character codes.

Appendix B: Flowchart Symbols

This appendix shows the flowchart symbols that are used in this book.

Appendix C: Pseudocode Reference

This appendix provides a quick reference for the pseudocode language that is used in the book.

Appendix D: Converting Decimal Numbers to Binary

This appendix uses a simple tutorial to demonstrate how to convert a decimal number to binary.

Appendix E: Answers to Checkpoint Questions

This appendix provides answers to the Checkpoint questions that appear throughout the text.

Organization of the Text

The text teaches programming logic and design in a step-by-step manner. Each chapter covers a major set of topics and builds knowledge as students progress through the book. Although the chapters can be easily taught in their existing sequence, there is some flexibility. Figure P-1 shows chapter dependencies. Each box represents a chapter or a group of chapters. A chapter to which an arrow points must be covered before the chapter from which the arrow originates. The dotted line indicates that only a portion of Chapter 10 depends on information presented in Chapter 8.

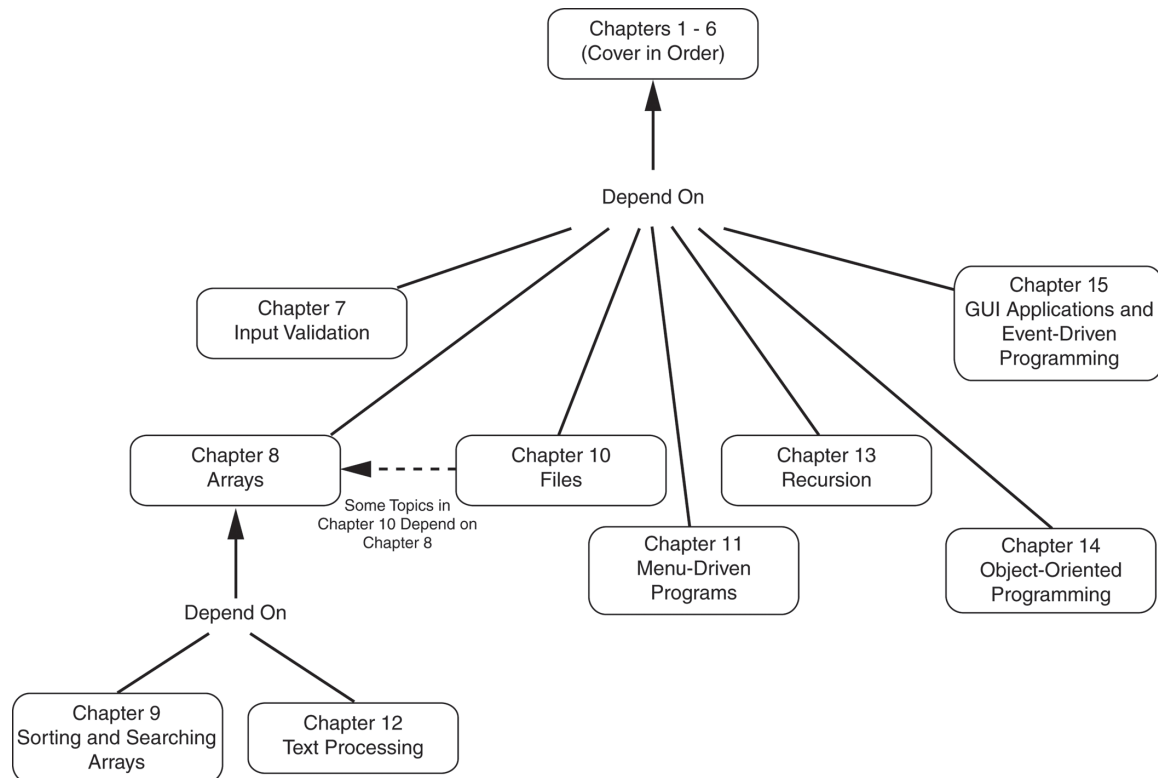
Features of the Text

Concept Statements. Each major section of the text starts with a concept statement. This statement concisely summarizes the main point of the section.

Example Programs. Each chapter has an abundant number of complete and partial example programs, each designed to highlight the current topic. Pseudocode, flowcharts, and other design tools are used in the example programs.

In the Spotlight. Each chapter has one or more *In the Spotlight* case studies that provide detailed, step-by-step analysis of problems, and show the student how to solve them.



Figure P-1 Chapter dependencies

VideoNotes. A series of online videos, developed specifically for this book, are available for viewing at www.pearson.com/cs-resources. Icons appear throughout the text alerting the student to videos about specific topics.



NOTE: Notes appear at several places throughout the text. They are short explanations of interesting or often misunderstood points relevant to the topic at hand.



TIP: Tips advise the student on the best techniques for approaching different programming or animation problems.



WARNING! Warnings caution students about programming techniques or practices that can lead to malfunctioning programs or lost data.



Programming Language Companions. Many of the pseudocode programs shown in this book have also been written in Java, Python, and Visual Basic. These programs appear in the programming language companions that are available at www.pearson.com/cs-resources. Icons appear next to each pseudocode program that also appears in the language companions.



Checkpoints. Checkpoints are questions placed at intervals throughout each chapter. They are designed to query the student's knowledge quickly after learning a new topic.

Review Questions. Each chapter presents a thorough and diverse set of Review Questions and exercises. They include Multiple Choice, True/False, Short Answer, and Algorithm Workbench.

Debugging Exercises. Most chapters provide a set of debugging exercises in which the student examines a set of pseudocode algorithms and identifies logical errors.

Programming Exercises. Each chapter offers a pool of Programming Exercises designed to solidify the student's knowledge of the topics currently being studied.

Supplements

Student Online Resources

Many student resources are available for this book from the publisher. The following items are available on the Gaddis Series resource page at www.pearson.com/cs-resources:

- **Access to the book's companion VideoNotes**

An extensive series of online VideoNotes have been developed to accompany this text. Throughout the book, VideoNote icons alert the student to videos covering specific topics. Additionally, one programming exercise at the end of each chapter has an accompanying VideoNote explaining how to develop the problem's solution.

- **Access to the Language Companions for Python, Java, Visual Basic, and C++**

Programming language companions specifically designed to accompany the Fourth Edition of this textbook are available for download. The companions introduce the Java™, Python®, Visual Basic®, and C++ programming languages, and correspond on a chapter-by-chapter basis with the textbook. Many of the pseudocode programs that appear in the textbook also appear in the companions, implemented in a specific programming language.

- **A link to download the Flowgorithm flowcharting application**

Flowgorithm is a free application, developed by Devin Cook at Sacramento State University, which allows you to create programs using simple flowcharts. It supports the flowcharting conventions used in this textbook, as well as several other standard conventions. When you create a flowchart with Flowgorithm, you can execute the program and generate Gaddis Pseudocode. You can also generate source code in Java, Python, Visual Basic, C#, Ruby, JavaScript, and several other languages. For more information, see www.flowgorithm.org.

- **A link to download the RAPTOR flowcharting environment**

RAPTOR is a flowchart-based programming environment developed by the US Air Force Academy Department of Computer Science.

Instructor Resources

The following supplements are available to qualified instructors only:

- Answers to all of the Review Questions
- Solutions for the Programming Exercises
- PowerPoint® presentation slides for each chapter
- Test bank

Visit the Pearson Instructor Resource Center www.pearson.com for information on how to access them.

Acknowledgments

There have been many helping hands in the development and publication of this text. I would like to thank the following faculty reviewers:

Reviewers for This Edition

Tony Cantrell
Georgia Northwestern Technical College

Keith Hallmark
Calhoun Community College

Vai Kumar
Pensacola State College

Reviewers of Previous Editions

Reni Abraham
Houston Community College

Alan Anderson
Gwinnett Technical College

Cherie Aukland
Thomas Nelson Community College

Steve Browning
Freed Hardeman University

John P. Buerck
Saint Louis University

Jill Canine
Ivy Tech Community College of Indiana

Steven D. Carver
Ivy Tech Community College

Stephen Robert Cheskiewicz
Keystone College and Wilkes University

Katie Danko
Grand Rapids Community College

Richard J. Davison
College of the Albemarle

Sameer Dutta
Grambling State University

Norman P. Hahn
Thomas Nelson Community College

John Haley
Athens Technical College

Ronald J. Harkins
Miami University, OH

Dianne Hill
Jackson College

Coronica Oliver
Coastal Georgia Community College

Robert S. Overall, III
Nashville State Community College

Dale T. Pickett
Baker College of Clinton Township

Tonya Pierce
Ivy Tech Community College

J. Shawn Pope
Tulsa Community College

Maryam Rahnemoonfar
Texas A&M University

Linda Reeser
Arizona Western College

Homayoun Sharafi
Prince George's Community College

Emily Shepard
Central Carolina Community College

Larry Strain
Ivy Tech Community College–Bloomington

Donald Stroup
Ivy Tech Community College

John Thacher
Gwinnett Technical College

Jim Turney
Austin Community College

Scott Vanselow
Edison State College

I also want to thank everyone at Pearson for making the *Starting Out With . . .* series so successful. I have worked so closely with the team at Pearson that I consider them among my closest friends. I am extremely fortunate to have Matt Goldstein as my editor, and Meghan Jacoby as Editorial Assistant. They have guided me through the process of revising this, and many other books. I am also fortunate to have Demetrius Hall as my marketing manager. His hard work is truly inspiring, and he does a great job getting my books out to the academic community. The production team, led by Carole Snyder, worked tirelessly to make this book a reality. Thanks to you all!

About the Author

Tony Gaddis is the principal author of the *Starting Out With . . .* series of textbooks. Tony has twenty years of experience teaching computer science courses, primarily at Haywood Community College. 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. The *Starting Out With . . .* series includes introductory books covering Programming Logic and Design, C++, Java, Microsoft® Visual Basic, C#®, Python, App Inventor, and Alice, all published by Pearson.

Fifth
Edition

Starting Out with



Programming Logic & Design

