

STARTING OUT WITH

JAVA™

From Control Structures
through Objects



STARTING OUT WITH

JAVA™

From Control Structures
through Objects

SEVENTH EDITION

Tony Gaddis

Haywood Community College



Pearson

330 Hudson Street, NY 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: Meghan Jacoby
Managing Content Producer: Scott Disanno
Content Producer: Amanda Brands
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
Cover Photo: Shutterstock/Tim UR
Printer/Binder: LSC Communications, Inc.
Full-Service Project Management: Sasibalan Chidambaram,
SPi Global

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

Copyright © 2019, 2016, 2013 by Pearson Education, Inc., Hoboken, New Jersey 07030. All rights reserved. Manufactured in the United States of America. This publication is protected by copyright and permissions 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 likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit <http://www.pearsoned.com/permissions/>.

Many of the designations by manufacturers and seller 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.

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.

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 Java. From control structures through objects / Tony Gaddis, Haywood Community College.

Description: Seventh edition. | NY, NY : Pearson Education, Inc., [2019] | Includes index.

Identifiers: LCCN 2017060354 | ISBN 9780134802213 | ISBN 0134802217

Subjects: LCSH: Java (Computer program language) | Data structures (Computer science) | Object-oriented programming (Computer science)

Classification: LCC QA76.73.J38 G333 2019 | DDC 005.13/3--dc23 LC record available at <https://lcn.loc.gov/2017060354>

1 18



ISBN 10: 0-13-480221-7
ISBN 13: 978-0-13-480221-3

Contents in Brief

Chapter 1	Introduction to Computers and Java	1
Chapter 2	Java Fundamentals	27
Chapter 3	Decision Structures	111
Chapter 4	Loops and Files	189
Chapter 5	Methods	269
Chapter 6	A First Look at Classes	317
Chapter 7	Arrays and the ArrayList Class	403
Chapter 8	A Second Look at Classes and Objects	493
Chapter 9	Text Processing and More about Wrapper Classes	557
Chapter 10	Inheritance	611
Chapter 11	Exceptions and Advanced File I/O	701
Chapter 12	JavaFX: GUI Programming and Basic Controls	759
Chapter 13	JavaFX: Advanced Controls	823
Chapter 14	JavaFX: Graphics, Effects, and Media	909
Chapter 15	Recursion	999
Chapter 16	Databases	1027
	Index	1109
	Appendices A–M	Companion Website
	Case Studies 1–7	Companion Website
	Chapters 17–20	Companion Website



Contents

Preface xxvi

Chapter 1 Introduction to Computers and Java 1

1.1	Introduction	1
1.2	Why Program?	1
1.3	Computer Systems: Hardware and Software	2
	<i>Hardware</i>	2
	<i>Software</i>	5
1.4	Programming Languages	6
	<i>What Is a Program?</i>	6
	<i>A History of Java</i>	8
1.5	What Is a Program Made Of?	8
	<i>Language Elements</i>	8
	<i>Lines and Statements</i>	11
	<i>Variables</i>	11
	<i>The Compiler and the Java Virtual Machine</i>	12
	<i>Java Software Editions</i>	13
	<i>Compiling and Running a Java Program</i>	14
1.6	The Programming Process	16
	<i>Software Engineering</i>	18
1.7	Object-Oriented Programming	19
	<i>Review Questions and Exercises</i>	21
	<i>Programming Challenge</i>	25

Chapter 2 Java Fundamentals 27

2.1	The Parts of a Java Program	27
2.2	The print and println Methods, and the Java API	33
2.3	Variables and Literals	39
	<i>Displaying Multiple Items with the + Operator</i>	40
	<i>Be Careful with Quotation Marks</i>	41
	<i>More about Literals</i>	42
	<i>Identifiers</i>	42
	<i>Class Names</i>	44

- 2.4 Primitive Data Types 44
 - The Integer Data Types* 46
 - Floating-Point Data Types* 47
 - The boolean Data Type* 50
 - The char Data Type* 50
 - Variable Assignment and Initialization* 52
 - Variables Hold Only One Value at a Time* 53
- 2.5 Arithmetic Operators 54
 - Integer Division* 57
 - Operator Precedence* 57
 - Grouping with Parentheses* 59
 - The Math Class* 62
- 2.6 Combined Assignment Operators 63
- 2.7 Conversion between Primitive Data Types 65
 - Mixed Integer Operations* 67
 - Other Mixed Mathematical Expressions* 68
- 2.8 Creating Named Constants with `final` 69
- 2.9 The String Class 70
 - Objects Are Created from Classes* 71
 - The String Class* 71
 - Primitive Type Variables and Class Type Variables* 71
 - Creating a String Object* 72
- 2.10 Scope 76
- 2.11 Comments 78
- 2.12 Programming Style 83
- 2.13 Reading Keyboard Input 85
 - Reading a Character* 89
 - Mixing Calls to `nextLine` with Calls to Other Scanner Methods* 89
- 2.14 Dialog Boxes 93
 - Displaying Message Dialogs* 93
 - Displaying Input Dialogs* 94
 - An Example Program* 94
 - Converting String Input to Numbers* 96
- 2.15 Common Errors to Avoid 99
 - Review Questions and Exercises* 100
 - Programming Challenges* 106

Chapter 3 **Decision Structures** 111

- 3.1 The `if` Statement 111
 - Using Relational Operators to Form Conditions* 113
 - Putting It All Together* 114
 - Programming Style and the `if` Statement* 117
 - Be Careful with Semicolons* 117
 - Having Multiple Conditionally Executed Statements* 118
 - Flags* 118

	<i>Comparing Characters</i>	119
3.2	The <code>if-else</code> Statement	120
3.3	Nested <code>if</code> Statements	122
3.4	The <code>if-else-if</code> Statement	128
3.5	Logical Operators	134
	<i>The Precedence of Logical Operators</i>	139
	<i>Checking Numeric Ranges with Logical Operators</i>	140
3.6	Comparing String Objects	142
	<i>Ignoring Case in String Comparisons</i>	146
3.7	More about Variable Declaration and Scope	147
3.8	The Conditional Operator (Optional)	149
3.9	The <code>switch</code> Statement	150
3.10	Displaying Formatted Output with <code>System.out.printf</code> and <code>String.format</code>	160
	<i>Format Specifier Syntax</i>	163
	<i>Precision</i>	164
	<i>Specifying a Minimum Field Width</i>	164
	<i>Flags</i>	167
	<i>Formatting String Arguments</i>	170
	<i>The <code>String.format</code> Method</i>	172
3.11	Common Errors to Avoid	174
	<i>Review Questions and Exercises</i>	175
	<i>Programming Challenges</i>	181

Chapter 4 **Loops and Files** 189

4.1	The Increment and Decrement Operators	189
	<i>The Difference between Postfix and Prefix Modes</i>	192
4.2	The <code>while</code> Loop	193
	<i>The <code>while</code> Loop Is a Pretest Loop</i>	196
	<i>Infinite Loops</i>	196
	<i>Don't Forget the Braces with a Block of Statements</i>	197
	<i>Programming Style and the <code>while</code> Loop</i>	198
4.3	Using the <code>while</code> Loop for Input Validation	200
4.4	The <code>do-while</code> Loop	204
4.5	The <code>for</code> Loop	207
	<i>The <code>for</code> Loop Is a Pretest Loop</i>	210
	<i>Avoid Modifying the Control Variable in the Body of the <code>for</code> Loop</i>	211
	<i>Other Forms of the Update Expression</i>	211
	<i>Declaring a Variable in the <code>for</code> Loop's Initialization Expression</i>	211
	<i>Creating a User Controlled <code>for</code> Loop</i>	212
	<i>Using Multiple Statements in the Initialization and Update Expressions</i>	213
4.6	Running Totals and Sentinel Values	216
	<i>Using a Sentinel Value</i>	219
4.7	Nested Loops	221
4.8	The <code>break</code> and <code>continue</code> Statements (Optional)	229

4.9 Deciding Which Loop to Use 229

4.10 Introduction to File Input and Output. 230

Using the `PrintWriter` Class to Write Data to a File 230

Appending Data to a File. 236

Specifying the File Location 237

Reading Data from a File. 237

Reading Lines from a File with the `nextLine` Method 238

Adding a `throws` Clause to the Method Header. 241

Checking for a File’s Existence 245

4.11 Generating Random Numbers with the `Random` Class. 249

4.12 Common Errors to Avoid 255

Review Questions and Exercises 256

Programming Challenges 262

Chapter 5 Methods 269

5.1 Introduction to Methods 269

void Methods and Value-Returning Methods 270

Defining a void Method 271

Calling a Method. 272

Layered Method Calls 276

Using Documentation Comments with Methods. 277

5.2 Passing Arguments to a Method. 279

Argument and Parameter Data Type Compatibility 281

Parameter Variable Scope 282

Passing Multiple Arguments 282

Arguments Are Passed by Value. 284

Passing Object References to a Method 285

Using the `@param` Tag in Documentation Comments 288

5.3 More about Local Variables 290

Local Variable Lifetime. 292

Initializing Local Variables with Parameter Values. 292

5.4 Returning a Value from a Method 293

Defining a Value-Returning Method 293

Calling a Value-Returning Method 294

Using the `@return` Tag in Documentation Comments 296

Returning a `boolean` Value. 300

Returning a Reference to an Object 300

5.5 Problem Solving with Methods 302

Calling Methods That Throw Exceptions 305

5.6 Common Errors to Avoid 305

Review Questions and Exercises 306

Programming Challenges 311

Chapter 6	A First Look at Classes	317
6.1	Objects and Classes	317
	<i>Classes: Where Objects Come From</i>	318
	<i>Classes in the Java API</i>	319
	<i>Primitive Variables vs. Objects</i>	321
6.2	Writing a Simple Class, Step by Step	324
	<i>Accessor and Mutator Methods</i>	338
	<i>The Importance of Data Hiding</i>	338
	<i>Avoiding Stale Data</i>	339
	<i>Showing Access Specification in UML Diagrams</i>	339
	<i>Data Type and Parameter Notation in UML Diagrams</i>	339
	<i>Layout of Class Members</i>	340
6.3	Instance Fields and Methods	341
6.4	Constructors	346
	<i>Showing Constructors in a UML Diagram</i>	348
	<i>Uninitialized Local Reference Variables</i>	348
	<i>The Default Constructor</i>	348
	<i>Writing Your Own No-Arg Constructor</i>	349
	<i>The String Class Constructor</i>	350
6.5	Passing Objects as Arguments	358
6.6	Overloading Methods and Constructors	370
	<i>The BankAccount Class</i>	372
	<i>Overloaded Methods Make Classes More Useful</i>	378
6.7	Scope of Instance Fields	378
	<i>Shadowing</i>	379
6.8	Packages and import Statements	380
	<i>Explicit and Wildcard import Statements</i>	380
	<i>The java.lang Package</i>	381
	<i>Other API Packages</i>	381
6.9	Focus on Object-Oriented Design: Finding the Classes and Their Responsibilities	382
	<i>Finding the Classes</i>	382
	<i>Identifying a Class's Responsibilities</i>	385
	<i>This Is Only the Beginning</i>	388
6.10	Common Errors to Avoid	388
	<i>Review Questions and Exercises</i>	389
	<i>Programming Challenges</i>	394
Chapter 7	Arrays and the ArrayList Class	403
7.1	Introduction to Arrays	403
	<i>Accessing Array Elements</i>	405

- Inputting and Outputting Array Contents* 406
- Java Performs Bounds Checking* 409
- Watch Out for Off-by-One Errors* 410
- Array Initialization* 411
- Alternate Array Declaration Notation* 412
- 7.2 Processing Array Elements 413
 - Array Length* 415
 - The Enhanced for Loop* 416
 - Letting the User Specify an Array's Size* 417
 - Reassigning Array Reference Variables* 419
 - Copying Arrays* 420
- 7.3 Passing Arrays as Arguments to Methods 422
- 7.4 Some Useful Array Algorithms and Operations 426
 - Comparing Arrays* 426
 - Summing the Values in a Numeric Array* 427
 - Getting the Average of the Values in a Numeric Array* 428
 - Finding the Highest and Lowest Values in a Numeric Array* 428
 - The SalesData Class* 429
 - Partially Filled Arrays* 437
 - Working with Arrays and Files* 438
- 7.5 Returning Arrays from Methods 439
- 7.6 String Arrays 441
 - Calling String Methods from an Array Element* 443
- 7.7 Arrays of Objects 444
- 7.8 The Sequential Search Algorithm 447
- 7.9 Two-Dimensional Arrays 450
 - Initializing a Two-Dimensional Array* 454
 - The Length Field in a Two-Dimensional Array* 455
 - Displaying All the Elements of a Two-Dimensional Array* 457
 - Summing All the Elements of a Two-Dimensional Array* 457
 - Summing the Rows of a Two-Dimensional Array* 458
 - Summing the Columns of a Two-Dimensional Array* 458
 - Passing Two-Dimensional Arrays to Methods* 459
 - Ragged Arrays* 461
- 7.10 Arrays with Three or More Dimensions 462
- 7.11 The Selection Sort and the Binary Search Algorithms 463
 - The Selection Sort Algorithm* 463
 - The Binary Search Algorithm* 466
- 7.12 Command-Line Arguments and Variable-Length Argument Lists 468
 - Command-Line Arguments* 469
 - Variable-Length Argument Lists* 470
- 7.13 The ArrayList Class 472
 - Creating and Using an ArrayList Object* 473
 - Using the Enhanced for Loop with an ArrayList* 474
 - The ArrayList Class's toString method* 475
 - Removing an Item from an ArrayList* 476

Inserting an Item 477
Replacing an Item 478
Capacity 479
Storing Your Own Objects in an ArrayList 479
Using the Diamond Operator for Type Inference 480
7.14 Common Errors to Avoid 481
Review Questions and Exercises 481
Programming Challenges 486

Chapter 8 A Second Look at Classes and Objects 493

8.1 Static Class Members 493
A Quick Review of Instance Fields and Instance Methods 493
Static Members 494
Static Fields 494
Static Methods 497
8.2 Passing Objects as Arguments to Methods 500
8.3 Returning Objects from Methods 503
8.4 The toString Method 505
8.5 Writing an equals Method 509
8.6 Methods That Copy Objects 512
Copy Constructors 514
8.7 Aggregation 515
Aggregation in UML Diagrams 523
Security Issues with Aggregate Classes 523
Avoid Using null References 525
8.8 The this Reference Variable 528
Using this to Overcome Shadowing 529
Using this to Call an Overloaded Constructor from Another Constructor . . 530
8.9 Enumerated Types 531
Enumerated Types Are Specialized Classes 532
Switching On an Enumerated Type 538
8.10 Garbage Collection 540
The finalize Method 542
8.11 Focus on Object-Oriented Design: Class Collaboration 542
Determining Class Collaborations with CRC Cards 545
8.12 Common Errors to Avoid 546
Review Questions and Exercises 547
Programming Challenges 551

Chapter 9 Text Processing and More about Wrapper Classes 557

9.1 Introduction to Wrapper Classes 557
9.2 Character Testing and Conversion with the Character Class 558
Character Case Conversion 563
9.3 More String Methods 566

Searching for Substrings 566
Extracting Substrings 572
Methods That Return a Modified String 576
The Static valueOf Methods 577

9.4 The *StringBuilder* Class 579
The StringBuilder Constructors 580
Other StringBuilder Methods 581
The toString Method 584

9.5 Tokenizing Strings 589

9.6 Wrapper Classes for the Numeric Data Types 594
The Static toString Methods 594
The toBinaryString, toHexString, and toOctalString Methods 594
The MIN_VALUE and MAX_VALUE Constants 595
Autoboxing and Unboxing 595

9.7 Focus on Problem Solving: The *TestScoreReader* Class 597

9.8 Common Errors to Avoid 601

Review Questions and Exercises 601
Programming Challenges 605

Chapter 10 Inheritance 611

10.1 What Is Inheritance? 611
Generalization and Specialization 611
Inheritance and the “Is a” Relationship 612
Inheritance in UML Diagrams 620
The Superclass’s Constructor 621
Inheritance Does Not Work in Reverse 623

10.2 Calling the Superclass Constructor 624
When the Superclass Has No Default or No-Arg Constructors 630
Summary of Constructor Issues in Inheritance 631

10.3 Overriding Superclass Methods 632
Overloading versus Overriding 637
Preventing a Method from Being Overridden 640

10.4 Protected Members 641
Package Access 646

10.5 Chains of Inheritance 647
Class Hierarchies 653

10.6 The *Object* Class 653

10.7 Polymorphism 655
Polymorphism and Dynamic Binding 656
The “Is-a” Relationship Does Not Work in Reverse 658
The instanceof Operator 659

10.8 Abstract Classes and Abstract Methods 660
Abstract Classes in UML 666

10.9 Interfaces 667
An Interface is a Contract 669

Fields in Interfaces 673
Implementing Multiple Interfaces 673
Interfaces in UML 673
Default Methods 674
Polymorphism and Interfaces 676
10.10 Anonymous Inner Classes 681
10.11 Functional Interfaces and Lambda Expressions 684
10.12 Common Errors to Avoid 689
Review Questions and Exercises 690
Programming Challenges 696

Chapter 11 Exceptions and Advanced File I/O 701

11.1 Handling Exceptions 701
Exception Classes 702
Handling an Exception 703
Retrieving the Default Error Message 707
Polymorphic References to Exceptions 710
Using Multiple catch Clauses to Handle Multiple Exceptions 710
The finally Clause 718
The Stack Trace 720
Handling Multiple Exceptions with One catch Clause 721
When an Exception Is Not Caught 723
Checked and Unchecked Exceptions 724
11.2 Throwing Exceptions 725
Creating Your Own Exception Classes 728
Using the @exception Tag in Documentation Comments 731
11.3 Advanced Topics: Binary Files, Random Access Files, and
Object Serialization 732
Binary Files 732
Random Access Files 739
Object Serialization 744
Serializing Aggregate Objects 748
11.4 Common Errors to Avoid 749
Review Questions and Exercises 749
Programming Challenges 755

Chapter 12 JavaFX: GUI Programming and Basic Controls 759

12.1 Graphical User Interfaces 759
Event-Driven GUI Programs 761
12.2 Introduction to JavaFX 762
Controls 762
Stages and Scenes 763
The Application Class 763
12.3 Creating Scenes 765

- Creating Controls 766
 - Creating Layout Containers 766
 - Creating a Scene Object 767
 - Adding the Scene Object to the Stage 768
 - Setting the Size of the Scene 770
 - Aligning Controls in an HBox Layout Container 770
- 12.4 Displaying Images 772
 - Loading Images from an Internet Location 775
 - Setting the Size of an Image 775
 - Preserving the Image’s Aspect Ratio 775
 - Changing an ImageView’s Image 776
- 12.5 More about the HBox, VBox, and GridPane Layout Containers. 776
 - The HBox Layout Container 777
 - The VBox Layout Container 782
 - The GridPane Layout Container 784
 - Using Multiple Layout Containers in the Same Screen 791
- 12.6 Button Controls and Events 792
 - Handling Events 794
 - Writing Event Handlers 794
 - Registering an Event Handler 795
- 12.7 Reading Input with TextField Controls 799
- 12.8 Using Anonymous Inner Classes and Lambda Expressions
 - to Handle Events 803
 - Using Anonymous Inner Classes to Create Event Handlers 803
 - Using Lambda Expressions to Create Event Handlers 806
- 12.9 The BorderPane Layout Container 808
- 12.10 The ObservableList Interface 812
- 12.11 Common Errors to Avoid 814
- Review Questions and Exercises 814
- Programming Challenges 818

Chapter 13 **JavaFX: Advanced Controls** 823

- 13.1 Styling JavaFX Applications with CSS 823
 - Type Selector Names 824
 - Style Properties 825
 - Applying a Stylesheet to a JavaFX Application 826
 - Applying Styles to the Root Node 830
 - Specifying Multiple Selectors in the Same Style Definition 832
 - Working with Colors 832
 - Creating a Custom Style Class Name 834
 - ID Selectors 836
 - Inline Style Rules 837
- 13.2 RadioButton Controls 838
 - Determining in Code Whether a RadioButton Is Selected 839
 - Selecting a RadioButton in Code 839

Responding to RadioButton Clicks 844

13.3 CheckBox Controls 848

Determining in Code Whether a CheckBox Is Selected 848

Selecting a CheckBox in Code 849

Responding to CheckBox Clicks 853

13.4 ListView Controls 853

Retrieving the Selected Item 855

Retrieving the Index of the Selected Item 856

Responding to Item Selection with an Event Handler 859

Adding Items versus Setting Items 860

Initializing a ListView with an Array or an ArrayList 861

Selection Modes 862

Retrieving Multiple Selected Items 863

Working With the Elements of an ObservableList 866

Converting an ObservableList to an Array 867

Using Code to Select an Item in a ListView 868

ListView Orientation 868

Creating ListViews of Objects Other Than String 869

13.5 ComboBox Controls 874

Retrieving the Selected Item 875

Responding to ComboBox Item Selection with an Event Handler 877

Editable ComboBoxes 879

13.6 Slider Controls 880

13.7 TextArea Controls 885

13.8 Menus 887

Assigning Mnemonics to Menu Items 895

13.9 The FileChooser Class 897

Displaying a FileChooser Dialog Box 898

13.10 Using Console Output to Debug a GUI Application 898

13.11 Common Errors to Avoid 902

Review Questions 902

Programming Challenges 906

Chapter 14 JavaFX: Graphics, Effects, and Media 909

14.1 Drawing Shapes 909

The Screen Coordinate System 909

The Shape Class and Its Subclasses 910

The Line Class 911

Changing the Stroke Color 914

The Circle Class 914

The Rectangle Class 918

The Ellipse Class 921

The Arc Class 924

The Polygon Class 928

The Polyline Class 931

The Text Class 933

Rotating Nodes 936

Scaling Nodes 938

14.2 Animation 940

The TranslateTransition Class 941

The RotateTransition Class 944

The ScaleTransition Class 949

The StrokeTransition Class 952

The FillTransition Class 953

The FadeTransition Class 954

Controlling the Animation 956

Specifying an Interpolator 956

14.3 Effects 958

The DropShadow Class 958

The InnerShadow Class 961

The ColorAdjust Class 962

The BoxBlur, GaussianBlur, and MotionBlur Classes 962

The SepiaTone Class 965

The Glow Class 966

The Reflection Class 967

Combining Effects 967

14.4 Playing Sound Files 969

Registering an EndOfMedia Event Handler 971

14.5 Playing Videos 974

14.6 Handling Key Events 979

Using an Anonymous Inner Class to Register a Key Event Handler
to the Scene 980

Using a Lambda Expression to Register a Key Event Handler
to the Scene 981

14.7 Handling Mouse Events 986

14.8 Common Errors to Avoid 992

Review Questions 992

Programming Challenges 995

Chapter 15 Recursion 999

15.1 Introduction to Recursion 999

15.2 Solving Problems with Recursion 1002

Direct and Indirect Recursion 1006

15.3 Examples of Recursive Methods 1007

Summing a Range of Array Elements with Recursion 1007

Drawing Concentric Circles 1008

The Fibonacci Series 1011

Finding the Greatest Common Divisor 1012

15.4 A Recursive Binary Search Method 1014

15.5 The Towers of Hanoi 1017
 15.6 Common Errors to Avoid 1021
Review Questions and Exercises 1022
Programming Challenges 1025

Chapter 16 Databases 1027

16.1 Introduction to Database Management Systems. 1027
 JDBC 1028
 SQL 1029
 Using a DBMS 1029
 Java DB 1030
 Creating the CoffeeDB Database 1030
 Connecting to the CoffeeDB Database 1030
 Connecting to a Password-Protected Database 1032
 16.2 Tables, Rows, and Columns 1033
 Column Data Types 1035
 Primary Keys 1035
 16.3 Introduction to the SQL SELECT Statement 1036
 Passing an SQL Statement to the DBMS 1038
 Specifying Search Criteria with the WHERE Clause 1048
 Sorting the Results of a SELECT Query 1054
 Mathematical Functions 1055
 16.4 Inserting Rows. 1058
 Inserting Rows with JDBC 1060
 16.5 Updating and Deleting Existing Rows. 1062
 Updating Rows with JDBC 1063
 Deleting Rows with the DELETE Statement 1067
 Deleting Rows with JDBC 1067
 16.6 Creating and Deleting Tables 1071
 Removing a Table with the DROP TABLE Statement 1074
 16.7 Creating a New Database with JDBC 1074
 16.8 Scrollable Result Sets 1076
 16.9 Result Set Metadata 1077
 16.10 Relational Data 1084
 Joining Data from Multiple Tables 1086
 An Order Entry System 1087
 16.11 Advanced Topics 1100
 Transactions 1100
 Stored Procedures 1101
 16.12 Common Errors to Avoid 1102
Review Questions and Exercises 1102
Programming Challenges 1107

Index 1109

The following appendices, online chapters, and online case studies are available on the book's online resource page at www.pearson.com/cs-resources.

Online Appendices:

- Appendix A: The ASCII/Unicode Characters
- Appendix B: Operator Precedence and Associativity
- Appendix C: Java Key Words
- Appendix D: Installing the JDK and Using the JDK Documentation
- Appendix E: Using the javadoc Utility
- Appendix F: More about the Math Class
- Appendix G: Packages
- Appendix H: Working with Records and Random - Access Files
- Appendix I: Configuring Java DB
- Appendix J: The QuickSort Algorithm
- Appendix K: Named Colors
- Appendix L: Answers to Checkpoints Questions
- Appendix M: Answers to Odd-Numbered Review Questions

Online Chapters:

- Chapter 17: GUI Applications with Swing – Part 1
- Chapter 18: GUI Applications with Swing – Part 2
- Chapter 19: Applets and More
- Chapter 20: Creating JavaFX Applications with Scene Builder

Online Case Studies:

- Case Study 1: Calculating Sales Commission
- Case Study 2: The Amortization Class
- Case Study 3: The PinTester Class
- Case Study 4: Parallel Arrays
- Case Study 5: The FeetInches Class
- Case Study 6: The SerialNumber Class
- Case Study 7: A Simple Text Editor Application

LOCATION OF VIDEONOTES IN THE TEXT



Chapter 1	Compiling and Running a Java Program, p. 14 Using an IDE, p. 14 Your First Java Program, p. 25
Chapter 2	Displaying Console Output, p. 33 Declaring Variables, p. 39 Simple Math Expressions, p. 55 The Miles-per-Gallon Problem, p. 107
Chapter 3	The if Statement, p. 111 The if-else Statement, p. 120 The if-else-if Statement, p. 129 The Time Calculator Problem, p. 182
Chapter 4	The while Loop, p. 193 The Pennies for Pay Problem, p. 263
Chapter 5	Passing Arguments to a Method, p. 279 Returning a Value from a Method, p. 293 The Retail Price Calculator Problem, p. 311
Chapter 6	Writing Classes and Creating Objects, p. 325 Initializing an Object with a Constructor, p. 346 The Personal Information Class Problem, p. 395
Chapter 7	Accessing Array Elements in a Loop, p. 407 Passing an Array to a Method, p. 422 The Charge Account Validation Problem, p. 487
Chapter 8	Returning Objects from Methods, p. 503 Aggregation, p. 515 The BankAccount, Class Copy Constructor Problem, p. 552
Chapter 9	The Sentence Capitalizer Problem, p. 605
Chapter 10	Inheritance, p. 611 Polymorphism, p. 655 The Employee and Productionworker Classes Problem, p. 696
Chapter 11	Handling Exceptions, p. 701 The Exception Project Problem, p. 757

(continued on the next page)

LOCATION OF VIDEONOTES IN THE TEXT *(continued)*



Chapter 12	Introduction to JavaFX, p. 762 Creating Scenes, p. 765 Displaying Images, p. 772 The HBox Layout Container, p. 777 The VBox Layout Container, p. 782 The GridPane Layout Container, p. 784 Button Controls and Events, p. 792 The TextField Control, p. 799 Using Anonymous Inner Classes as Event Handlers, p. 803 Using Lambda Expressions as Event Handlers, p. 806 The Latin Translator Problem, p. 818
Chapter 13	JavaFX and CSS, p. 823 RadioButton Controls, p. 838 CheckBox Controls, p. 848 ListView Controls, p. 853 ComboBox Controls, p. 874 Slider Controls, p. 880, The Dorm and Meal Plan Calculator Problem, p. 906
Chapter 14	Drawing Shapes with JavaFX, p. 909 JavaFX Animation, p. 940 JavaFX Effects, p. 958 Playing Sound Files with JavaFX, p. 970 Playing Videos with JavaFX, p. 974 Handling Key Events in JavaFX, p. 979 Handling Mouse Events in JavaFX, p. 986 The Coin Toss Problem, p. 996
Chapter 15	Reducing a Problem with Recursion, p. 1003 The Recursive Power Problem, p. 1026
Chapter 16	The Customer Inserter Problem, p. 1107

Preface

Welcome to *Starting Out with Java: From Control Structures through Objects*, Seventh Edition. This book is intended for a one-semester or a two-quarter CS1 course. Although it is written for students with no prior programming background, even experienced students will benefit from its depth of detail.

Control Structures First, Then Objects

This text first introduces the student to the fundamentals of data types, input and output, control structures, methods, and objects created from standard library classes.

Next, the student learns to use arrays of primitive types and reference types. After this, the student progresses through more advanced topics, such as inheritance, polymorphism, the creation and management of packages, GUI applications, recursion, and database programming. From early in the book, applications are documented with javadoc comments. As the student progresses through the text, new javadoc tags are covered and demonstrated.

As with all the books in the *Starting Out With ...* series, the hallmark of this text is its clear, friendly, and easy-to-understand writing. In addition, it is rich in example programs that are concise and practical.

Changes in the Seventh Edition

This book's pedagogy, organization, and clear writing style remain the same as in the previous edition. The most significant change in this edition is the switch from Swing to JavaFX in the chapters that focus on GUI development. Although Swing is not officially deprecated, Oracle has announced that JavaFX has replaced Swing as the standard GUI library for Java.¹

In this edition, we have added the following new chapters:

- **Chapter 12 JavaFX: GUI Programming and Basic Controls** This chapter presents the basics of developing graphical user interface (GUI) applications with JavaFX. Fundamental controls, layout containers, and the basic concepts of event-driven programming are covered.
- **Chapter 13 JavaFX: Advanced Controls** This chapter discusses CSS styling and advanced user interface controls.

¹ <http://www.oracle.com/technetwork/java/javafx/overview/faq-1446554.html#6>

- **Chapter 14 JavaFX: Graphics, Effects, and Media** This chapter discusses 2D shapes, animation, visual effects, playing audio and video, and responding to mouse and keyboard events.

The Swing and Applet material that appeared in the previous edition is still available on the book's companion Web site, as the following online chapters:

- The previous Chapter 12 *A First Look At GUI Applications* is now available online as Chapter 17.
- The previous Chapter 13 *Advanced GUI Applications* is now available online as Chapter 18.
- The previous Chapter 14 *Applets and More* is now available online as Chapter 19.

Note: Chapter 15 from the previous edition has also been moved to the book's companion Web site as *Chapter 20 Creating JavaFX Applications with Scene Builder*. Although Oracle no longer officially supports Scene Builder, it is still available as an open source tool at <http://gluonhq.com/labs/scene-builder/>

In addition to the new JavaFX chapters, the Database chapter, which is now Chapter 15, has been updated to use JavaFX instead of Swing for its GUI applications. We have also added several new, motivational programming problems throughout the book.

Organization of the Text

The text teaches Java step-by-step. 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. An arrow points from a chapter to the chapter that must be previously covered.

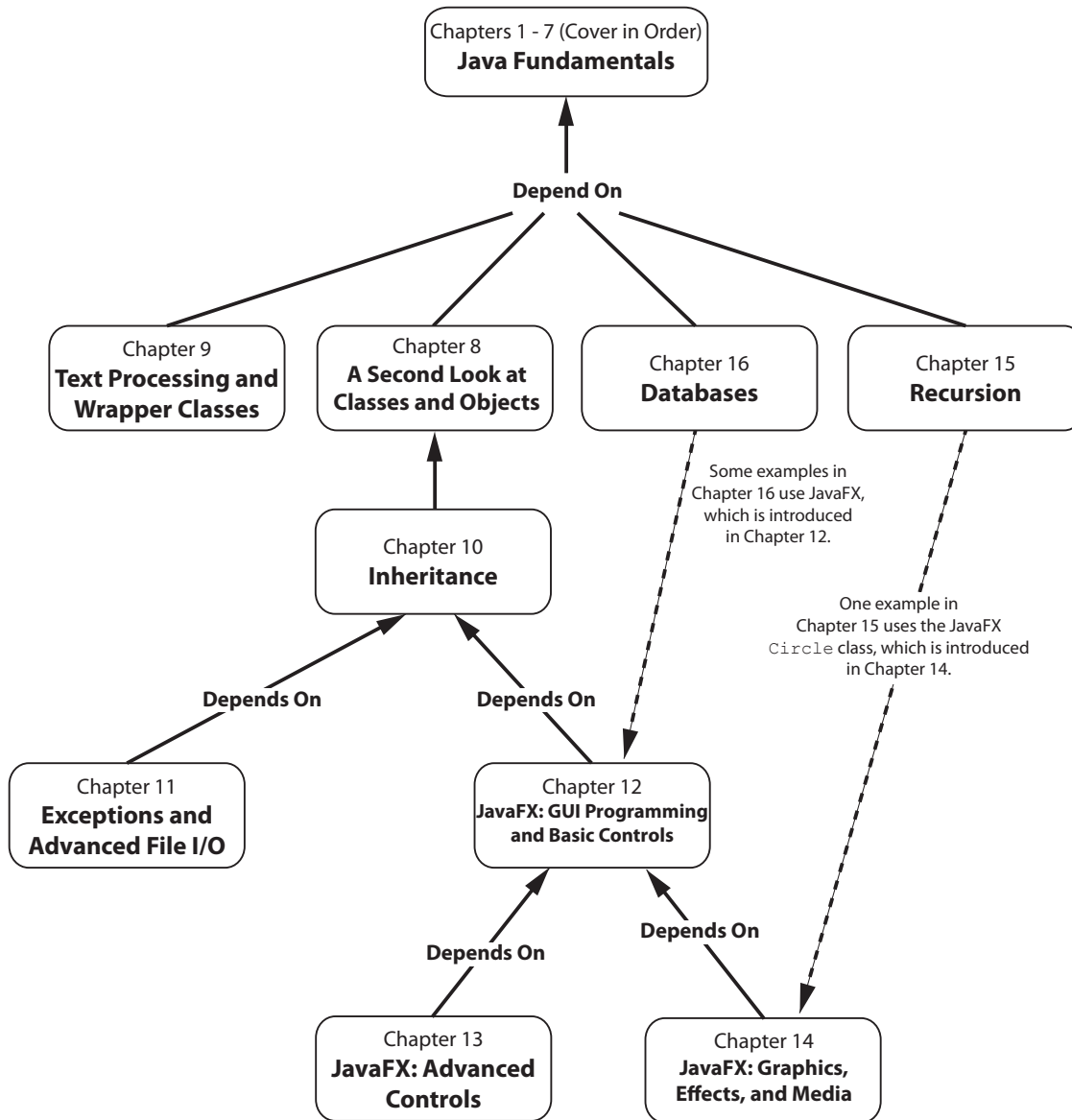
Brief Overview of Each Chapter

Chapter 1: Introduction to Computers and Java. This chapter provides an introduction to the field of computer science and covers the fundamentals of hardware, software, and programming languages. The elements of a program, such as key words, variables, operators, and punctuation, are discussed by examining a simple program. An overview of entering source code, compiling, and executing a program is presented. A brief history of Java is also given.

Chapter 2: Java Fundamentals. This chapter gets students started in Java by introducing data types, identifiers, variable declarations, constants, comments, program output, and simple arithmetic operations. The conventions of programming style are also introduced. Students learn to read console input with the `Scanner` class and with dialog boxes using `JOptionPane`.

Chapter 3: Decision Structures. In this chapter students explore relational operators and relational expressions and are shown how to control the flow of a program with the `if`, `if-else`, and `if-else-if` statements. Nested `if` statements, logical operators, the conditional operator, and the `switch` statement are also covered. The chapter discusses how to compare `String` objects with the `equals`, `compareTo`, `equalsIgnoreCase`, and `compareToIgnoreCase`

Figure P-1 Chapter dependencies



methods. Formatting numeric output with the `System.out.printf` method and the `String.format` method is discussed.

Chapter 4: Loops and Files. This chapter covers Java’s 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. Simple file operations for reading and writing text files are included.

Chapter 5: Methods. In this chapter students learn how to write void methods, value-returning methods, and methods that do and do not accept arguments. The concept of functional decomposition is discussed.

Chapter 6: A First Look at Classes. This chapter introduces students to designing classes for the purpose of instantiating objects. Students learn about class fields and methods, and UML diagrams are introduced as a design tool. Then constructors and overloading are discussed. A `BankAccount` class is presented as a case study, and a section on object-oriented design is included. This section leads the students through the process of identifying classes and their responsibilities within a problem domain. There is also a section that briefly explains packages and the `import` statement.

Chapter 7: Arrays and the `ArrayList` Class. In this chapter students learn to create and work with single and multi-dimensional arrays. Numerous array-processing techniques are demonstrated, such as summing the elements in an array, finding the highest and lowest values, and sequentially searching an array. Other topics, including ragged arrays and variable-length arguments (`varargs`), are also discussed. The `ArrayList` class is introduced, and Java's generic types are briefly discussed and demonstrated.

Chapter 8: A Second Look at Classes and Objects. This chapter shows students how to write classes with added capabilities. Static methods and fields, interaction between objects, passing objects as arguments, and returning objects from methods are discussed. Aggregation and the “has a” relationship is covered, as well as enumerated types. A section on object-oriented design shows how to use CRC cards to determine the collaborations among classes.

Chapter 9: Text Processing and More about Wrapper Classes. This chapter discusses the numeric and `Character` wrapper classes. Methods for converting numbers to strings, testing the case of characters, and converting the case of characters are covered. Autoboxing and unboxing are also discussed. More `String` class methods are covered, including using the `split` method to tokenize strings. The chapter also covers the `StringBuilder` class.

Chapter 10: Inheritance. The study of classes continues in this chapter with the subjects of inheritance and polymorphism. The topics covered include superclasses, subclasses, how constructors work in inheritance, method overriding, polymorphism and dynamic binding, protected and package access, class hierarchies, abstract classes, abstract methods, anonymous inner classes, interfaces, and lambda expressions.

Chapter 11: Exceptions and Advanced File I/O. In this chapter students learn to develop enhanced error trapping techniques using exceptions. Handling exceptions is covered, as well as developing and throwing custom exceptions. The chapter discusses advanced techniques for working with sequential access, random access, text, and binary files.

Chapter 12: JavaFX: GUI Programming and Basic Controls. This chapter presents the basics of developing graphical user interface (GUI) applications with JavaFX. Fundamental controls, layout containers, and the basic concepts of event-driven programming are covered.

Chapter 13: JavaFX: Advanced Controls. This chapter discusses CSS styling and advanced user interface controls, such as `RadioButtons`, `CheckBoxes`, `ListViews`, `ComboBoxes`, `Sliders`, and `TextAreas`. Menu systems and `FileChooser`s are also covered.

Chapter 14: JavaFX: Graphics, Effects, and Media. This chapter discusses 2D shapes, animation, visual effects, playing audio and video, and responding to mouse and keyboard events.

Chapter 15: Recursion. This chapter presents recursion as a problem-solving technique. Numerous examples of recursive methods are demonstrated.


Chapter 16: Databases. This chapter introduces the student to database programming. The basic concepts of database management systems and SQL are first introduced. Then the student learns to use JDBC to write database applications in Java. Relational data is covered, and numerous example programs are presented throughout the chapter.


Features of the Text


Concept Statements. Each major section of the text starts with a concept statement that concisely summarizes the focus of the section.

Example Programs. The text has an abundant number of complete and partial example programs, each designed to highlight the current topic. In most cases the programs are practical, real-world examples.

Program Output. Each example program is followed by a sample of its output, which shows students how the program functions.


 **Checkpoints.** Checkpoints, highlighted by the checkmark icon, appear at intervals throughout each chapter. They are designed to check students' knowledge soon after learning a new topic. Answers for all Checkpoint questions are provided in Appendix K, which can be downloaded from the book's resource page at www.pearson.com/cs-resources.

 **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 problems and appear regularly throughout the text.

 **WARNING!** Warnings caution students about certain Java features, programming techniques, or practices that can lead to malfunctioning programs or lost data.

In the Spotlight. Many of the chapters provide an *In the Spotlight* section that presents a programming problem, along with detailed, step-by-step analysis showing the student how to solve it.

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



Case Studies. Case studies that simulate real-world business applications are introduced throughout the text and are provided on the book’s resource page at www.pearson.com/cs-resources.

Common Errors to Avoid. Each chapter provides a list of common errors and explanations of how to avoid them.

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

Programming Challenges. Each chapter offers a pool of programming challenges designed to solidify students’ knowledge of topics at hand. In most cases the assignments present real-world problems to be solved.

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:

- The source code for each example program in the book
- Access to the book’s companion VideoNotes
- Appendixes A–M (listed in the Contents)
- A collection of seven valuable Case Studies (listed in the Contents)
- Online Chapters 17–20 (listed in the Contents)
- Links to download the Java™ Development Kit
- Links to download numerous programming environments including jGRASP™, Eclipse™, TextPad™, NetBeans™, JCreator, and DrJava

Online Practice and Assessment with MyLab Programming

MyLab Programming helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyLab Programming 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, the MyLab Programming 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.

MyLab Programming 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 MyLab Programming in your course, visit www.pearson.com/mylab/programming.

Instructor Resources

The following supplements are available to qualified instructors:

- Answers to all of the Review Questions
- Solutions for the Programming Challenges
- PowerPoint Presentation slides for each chapter
- Computerized Test Banks
- Source Code
- Lab Manual
- Student Files for the Lab Manual
- Solutions to the Lab Manual

Visit the Pearson Instructor Resource Center (www.pearson.com) or contact your local Pearson representative for information on how to access these resources.

Acknowledgments

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

Reviewers For This Edition

John Bono
George Mason University

Irene Bruno
George Mason University

Jackie Horton
University of Vermont

Mohammad T. Islam
Southern Connecticut State University

David Krebs
University of Pittsburgh

Reviewers of Previous Editions

Ahmad Abuhejleh
University of Wisconsin, River Falls

Colin Archibald
Valencia Community College

Ijaz Awani
Savannah State University

Bill Bane
Tarleton State University

N. Dwight Barnette
Virginia Tech

Asoke Bhattacharyya
Saint Xavier University, Chicago

Marvin Bishop
Manhattan College

Heather Booth
University of Tennessee, Knoxville

David Boyd
Valdosta University

Julius Brandstatter
Golden Gate University

Kim Cannon
Greenville Tech

Jesse Cecil
College of the Siskiyous

James Chegwidden
Tarrant County College

Kay Chen
Bucks County Community College

- Brad Chilton
Tarleton State University
- Diane Christie
University of Wisconsin, Stout
- Cara Cocking
Marquette University
- Jose Cordova
University of Louisiana, Monroe
- Walter C. Daugherty
Texas A & M University
- Michael Doherty
University of the Pacific
- Jeanne M. Douglas
University of Vermont
- Sander Eller
*California Polytechnic University,
Pomona*
- Brooke Estabrook-Fishinghawk
Mesa Community College
- Mike Fry
Lebanon Valley College
- David Goldschmidt
College of St. Rose
- Georgia R. Grant
College of San Mateo
- Carl Stephen Guynes
University of North Texas
- Nancy Harris
James Madison University
- Chris Haynes
Indiana University
- Ric Heishman
Northern Virginia Community College
- Deedee Herrera
Dodge City Community College
- Mary Hovik
Lehigh Carbon Community College
- Brian Howard
DePauw University
- Alan Jackson
Oakland Community College (MI)
- Norm Jacobson
University of California, Irvine
- Zhen Jiang
West Chester University
- Stephen Judd
University of Pennsylvania
- Neven Jurkovic
Palo Alto College
- Dennis Lang
Kansas State University
- Jiang Li
Austin Peay State University
- Harry Lichtbach
Evergreen Valley College
- Michael A. Long
California State University, Chico
- Cheng Luo
Coppin State University
- Tim Margush
University of Akron
- Blayne E. Mayfield
Oklahoma State University
- Scott McLeod
Riverside Community College
- Dean Mellas
Cerritos College
- Georges Merx
San Diego Mesa College
- Martin Meyers
*California State University,
Sacramento*
- Pati Milligan
Baylor University
- Laurie Murphy
Pacific Lutheran University
- Steve Newberry
Tarleton State University

Lynne O’Hanlon
Los Angeles Pierce College

Merrill Parker
Chattanooga State Technical Community College

Bryson R. Payne
North Georgia College and State University

Rodney Pearson
Mississippi State University

Peter John Polito
Springfield College

Charles Robert Putnam
California State University, Northridge

Y. B. Reddy
Grambling State University

Elizabeth Riley
Macon State College

Felix Rodriguez
Naugatuck Valley Community College

Diane Rudolph
John A Logan College

Carolyn Schauble
Colorado State University

Bonnie Smith
Fresno City College

Daniel Spiegel
Kutztown University

Caroline St. Clair
North Central College

Karen Stanton
Los Medanos College

Timothy Urness
Drake University

Peter van der Goes
Rose State College

Tuan A Vo
Mt. San Antonio College

Xiaoying Wang
University of Mississippi

Yu Wu
University of North Texas

Zijiang Yang
Western Michigan University

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 book, as well as many others. 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 Amanda Brands, 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. He has nearly two decades of experience teaching computer science courses, primarily at Haywood Community College. Tony 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 textbooks covering programming logic and design, C++, Java™, Microsoft® Visual Basic®, Microsoft® Visual C#, Python, Alice, and App Inventor, all published by Pearson.



MyLab Programming

Through the power of practice and immediate personalized feedback, MyLab™ Programming helps students master programming fundamentals and build computational thinking skills.

PROGRAMMING PRACTICE

With MyLab Programming, your students will gain first-hand programming experience in an interactive online environment.

IMMEDIATE, PERSONALIZED FEEDBACK

MyLab Programming automatically detects errors in the logic and syntax of their code submission and offers targeted hints that enables students to figure out what went wrong and why.

GRADUATED COMPLEXITY

MyLab Programming breaks down programming concepts into short, understandable sequences of exercises. Within each sequence the level and sophistication of the exercises increase gradually but steadily.

DYNAMIC ROSTER

Students' submissions are stored in a roster that indicates whether the submission is correct, how many attempts were made, and the actual code submissions from each attempt.

PEARSON eTEXT

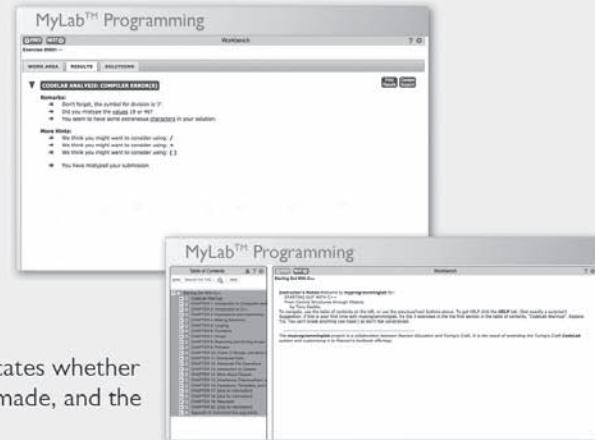
The Pearson eText gives students access to their textbook anytime, anywhere.

STEP-BY-STEP VIDEONOTE TUTORIALS

These step-by-step video tutorials enhance the programming concepts presented in select Pearson textbooks.

For more information and titles available with **MyLab Programming**, please visit www.pearson.com/mylab/programming

Copyright © 2018 Pearson Education, Inc. or its affiliate(s). All rights reserved. HELO88173 • 11/15



ALWAYS LEARNING



