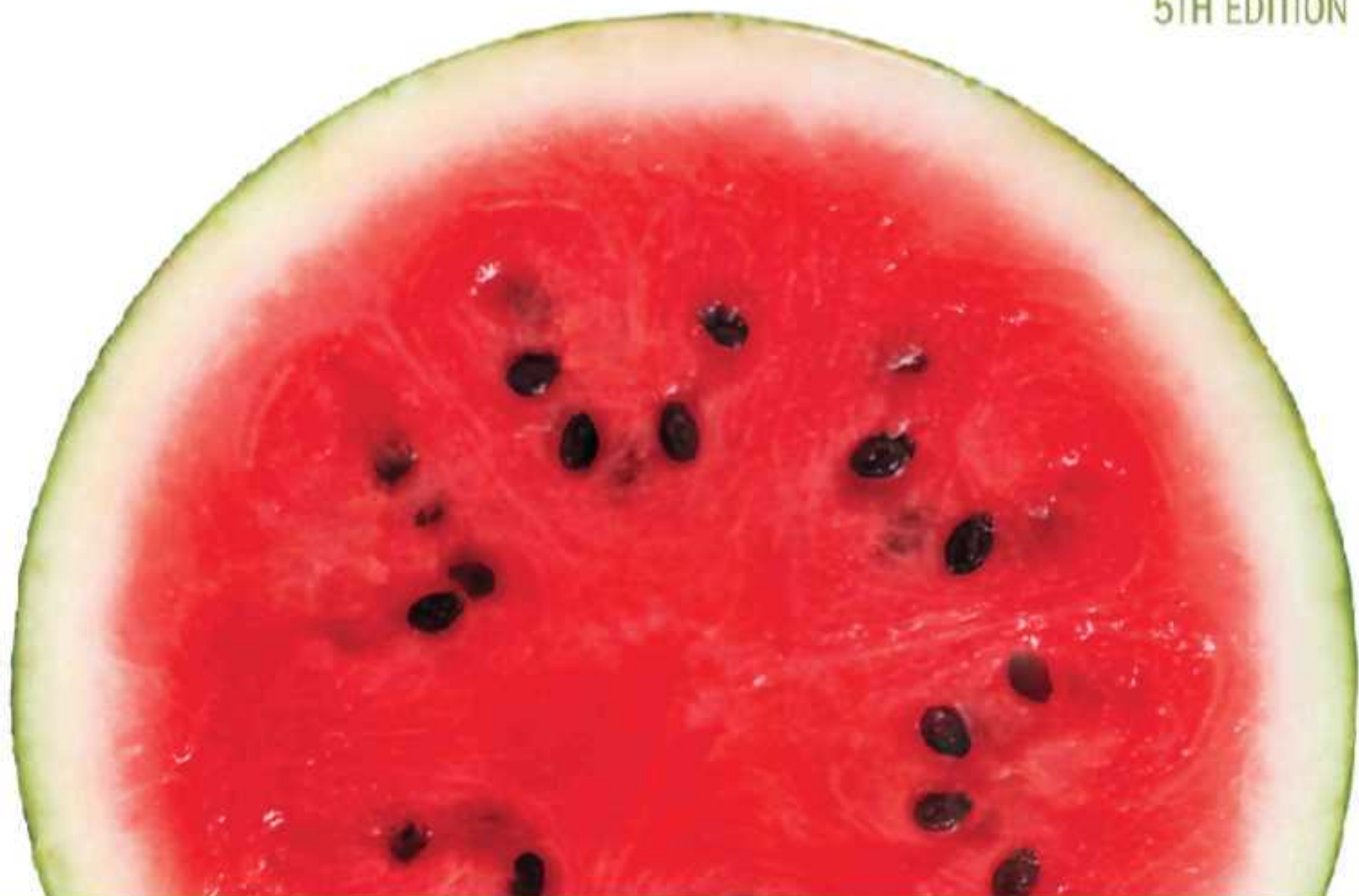


starting out with >>>

JAVA™

From Control Structures through Objects

5TH EDITION



TONY GADDIS



get with the programming

Through the power of practice and immediate personalized feedback, MyProgrammingLab improves your performance.

MyProgrammingLabTM

Learn more at www.myprogramminglab.com

ALWAYS LEARNING

PEARSON

STARTING OUT WITH

JAVA™

From Control Structures
through Objects

FIFTH EDITION

Tony Gaddis

Haywood Community College

PEARSON

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: Marcia Horton
Editor in Chief: Michael Hirsch
Acquisitions Editor: Matt Goldstein
Editorial Assistant: Chelsea Kharakozova
Director of Marketing: Patrice Jones
Marketing Manager: Yez Alayan
Marketing Coordinator: Kathryn Ferranti
Marketing Assistant: Emma Snider
Director of Production: Vince O'Brien
Managing Editor: Jeff Holcomb
Senior Production Project Manager: Marilyn Lloyd
Production Editor: Pat Brown

Manufacturing Buyer: Pat Brown
Art Director: Anthony Gemmellaro
Cover Designer: Joyce Cosentino Wells
Cover Image: © istockphoto
Media Project Manager: John Cassar
Text Permissions: Jenn Kennett/Creative Compliance
Full-Service Project Management: Jogender Taneja,
Aptara®, Inc.
Composition: Aptara®, Inc.
Printer/Bindery: Edwards Brothers
Cover printer: Lehigh-Phoenix Color/
Hagerstown

Copyright © 2013, 2010, 2008, 2005 by Pearson Education, Inc., publishing as Addison-Wesley. All rights reserved. Manufactured 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 likewise. To obtain permission(s) to use material from this work, please submit a written request to Pearson Education, Inc., Permissions Department, One Lake Street, Upper Saddle River, New Jersey 07458, or you may fax your request to 201-236-3290.

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.

Library of Congress Cataloging-in-Publication Data

Gaddis, Tony.

Starting out with Java : from control structures through objects / Tony Gaddis. — 5th ed.
p. cm.

ISBN-13: 978-0-13-285583-9

ISBN-10: 0-13-285583-6

1. Java (Computer program language) 2. Data structures (Computer science) I. Title.

QA76.73.J38G33 2013

005.13'3—dc23

2012000363

10 9 8 7 6 5 4 3 2 1

PEARSON

SVE ISBN-13: 978-0-13-289038-0

SVE ISBN-10: 0-13-289038-0

ISBN-13: 978-0-13-285583-9

ISBN-10: 0-13-285583-6

Contents in Brief

	Preface	xv	
Chapter 1	Introduction to Computers and Java	1	
Chapter 2	Java Fundamentals	27	
Chapter 3	Decision Structures	109	
Chapter 4	Loops and Files	193	
Chapter 5	Methods	273	
Chapter 6	A First Look at Classes	323	
Chapter 7	Arrays and the ArrayList Class	407	
Chapter 8	A Second Look at Classes and Objects	497	
Chapter 9	Text Processing and More about Wrapper Classes	561	
Chapter 10	Inheritance	619	
Chapter 11	Exceptions and Advanced File I/O	699	
Chapter 12	A First Look at GUI Applications	755	
Chapter 13	Advanced GUI Applications	843	
Chapter 14	Applets and More	911	
Chapter 15	Recursion	985	
Chapter 16	Databases	1013	
	Java™ Quick Reference	1109	
	Index	1111	
	Credits	1127	
	Appendixes A–M		Companion Website
	Case Studies 1–7		Companion Website

Contents

Preface xv

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
1.4	Programming Languages	6
1.5	What Is a Program Made Of?	9
1.6	The Programming Process	16
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 <code>print</code> and <code>println</code> Methods, and the Java API	33
2.3	Variables and Literals	39
2.4	Primitive Data Types	44
2.5	Arithmetic Operators	54
2.6	Combined Assignment Operators	63
2.7	Conversion between Primitive Data Types	65
2.8	Creating Named Constants with <code>final</code>	69
2.9	The <code>String</code> Class	70
2.10	Scope	75
2.11	Comments	77
2.12	Programming Style	82
2.13	Reading Keyboard Input	84
2.14	Dialog Boxes	92
2.15	Common Errors to Avoid	99
	<i>Review Questions and Exercises</i>	100
	<i>Programming Challenges</i>	105

Chapter 3 Decision Structures 109

3.1	The if Statement.....	109
3.2	The if-else Statement.....	119
3.3	Nested if Statements.....	122
3.4	The if-else-if Statement.....	129
3.5	Logical Operators.....	135
3.6	Comparing String Objects.....	143
3.7	More about Variable Declaration and Scope.....	149
3.8	The Conditional Operator (Optional).....	150
3.9	The switch Statement.....	152
3.10	The System.out.printf Method.....	162
3.11	Creating Objects with the DecimalFormat Class.....	174
3.12	Common Errors to Avoid.....	181
	<i>Review Questions and Exercises</i>	182
	<i>Programming Challenges</i>	187

Chapter 4 Loops and Files 193

4.1	The Increment and Decrement Operators.....	193
4.2	The while Loop.....	197
4.3	Using the while Loop for Input Validation.....	204
4.4	The do-while Loop.....	208
4.5	The for Loop.....	211
4.6	Running Totals and Sentinel Values.....	220
4.7	Nested Loops.....	225
4.8	The break and continue Statements (Optional).....	233
4.9	Deciding Which Loop to Use.....	233
4.10	Introduction to File Input and Output.....	234
4.11	Generating Random Numbers with the Random Class.....	253
4.12	Common Errors to Avoid.....	259
	<i>Review Questions and Exercises</i>	260
	<i>Programming Challenges</i>	266

Chapter 5 Methods 273

5.1	Introduction to Methods.....	273
5.2	Passing Arguments to a Method.....	283
5.3	More about Local Variables.....	295
5.4	Returning a Value from a Method.....	297
5.5	Problem Solving with Methods.....	306
5.6	Common Errors to Avoid.....	310
	<i>Review Questions and Exercises</i>	311
	<i>Programming Challenges</i>	316

Chapter 6 A First Look at Classes 323

6.1	Objects and Classes	323
6.2	Writing a Simple Class, Step by Step	330
6.3	Instance Fields and Methods	347
6.4	Constructors	352
6.5	Passing Objects as Arguments	364
6.6	Overloading Methods and Constructors	376
6.7	Scope of Instance Fields	384
6.8	Packages and <code>import</code> Statements	386
6.9	Focus on Object-Oriented Design: Finding the Classes and Their Responsibilities	388
6.10	Common Errors to Avoid	394
	<i>Review Questions and Exercises</i>	395
	<i>Programming Challenges</i>	400

Chapter 7 Arrays and the ArrayList Class 407

7.1	Introduction to Arrays	407
7.2	Processing Array Elements	417
7.3	Passing Arrays as Arguments to Methods	426
7.4	Some Useful Array Algorithms and Operations	430
7.5	Returning Arrays from Methods	443
7.6	<code>String</code> Arrays	445
7.7	Arrays of Objects	448
7.8	The Sequential Search Algorithm	451
7.9	Two-Dimensional Arrays	454
7.10	Arrays with Three or More Dimensions	466
7.11	The Selection Sort and the Binary Search Algorithms	467
7.12	Command-Line Arguments and Variable-Length Argument Lists	472
7.13	The <code>ArrayList</code> Class	476
7.14	Common Errors to Avoid	485
	<i>Review Questions and Exercises</i>	485
	<i>Programming Challenges</i>	490

Chapter 8 A Second Look at Classes and Objects 497

8.1	Static Class Members	497
8.2	Passing Objects as Arguments to Methods	504
8.3	Returning Objects from Methods	507
8.4	The <code>toString</code> Method	509
8.5	Writing an <code>equals</code> Method	513
8.6	Methods That Copy Objects	516
8.7	Aggregation	519
8.8	The <code>this</code> Reference Variable	532
8.9	Enumerated Types	535
8.10	Garbage Collection	544

8.11	Focus on Object-Oriented Design: Class Collaboration	546
8.12	Common Errors to Avoid	550
	<i>Review Questions and Exercises</i>	551
	<i>Programming Challenges</i>	555

Chapter 9 **Text Processing and More about Wrapper Classes** 561

9.1	Introduction to Wrapper Classes.	561
9.2	Character Testing and Conversion with the <code>Character</code> Class	562
9.3	More String Methods	570
9.4	The <code>StringBuilder</code> Class	584
9.5	Tokenizing Strings	595
9.6	Wrapper Classes for the Numeric Data Types	603
9.7	Focus on Problem Solving: The <code>TestScoreReader</code> Class	606
9.8	Common Errors to Avoid	610
	<i>Review Questions and Exercises</i>	610
	<i>Programming Challenges</i>	614

Chapter 10 **Inheritance** 619

10.1	What Is Inheritance?	619
10.2	Calling the Superclass Constructor	632
10.3	Overriding Superclass Methods	640
10.4	Protected Members	649
10.5	Chains of Inheritance	655
10.6	The <code>Object</code> Class	661
10.7	Polymorphism	663
10.8	Abstract Classes and Abstract Methods	668
10.9	Interfaces	675
10.10	Common Errors to Avoid	687
	<i>Review Questions and Exercises</i>	688
	<i>Programming Challenges</i>	693

Chapter 11 **Exceptions and Advanced File I/O** 699

11.1	Handling Exceptions	699
11.2	Throwing Exceptions	723
11.3	Advanced Topics: Binary Files, Random Access Files, and Object Serialization	729
11.4	Common Errors to Avoid	746
	<i>Review Questions and Exercises</i>	746
	<i>Programming Challenges</i>	752

Chapter 12 **A First Look at GUI Applications** 755

12.1	Introduction	755
12.2	Creating Windows	758

12.3	Layout Managers	787
12.4	Radio Buttons and Check Boxes	804
12.5	Borders	815
12.6	Focus on Problem Solving: Extending Classes from JPanel	818
12.7	Splash Screens	830
12.8	Using Console Output to Debug a GUI Application	831
12.9	Common Errors to Avoid	836
	<i>Review Questions and Exercises</i>	836
	<i>Programming Challenges</i>	839

Chapter 13 **Advanced GUI Applications** 843

13.1	The Swing and AWT Class Hierarchy	843
13.2	Read-Only Text Fields	844
13.3	Lists	846
13.4	Combo Boxes	862
13.5	Displaying Images in Labels and Buttons	868
13.6	Mnemonics and Tool Tips	874
13.7	File Choosers and Color Choosers	876
13.8	Menus	880
13.9	More about Text Components: Text Areas and Fonts	889
13.10	Sliders	893
13.11	Look and Feel	898
13.12	Common Errors to Avoid	900
	<i>Review Questions and Exercises</i>	901
	<i>Programming Challenges</i>	906

Chapter 14 **Applets and More** 911

14.1	Introduction to Applets	911
14.2	A Brief Introduction to HTML	913
14.3	Creating Applets with Swing	922
14.4	Using AWT for Portability	931
14.5	Drawing Shapes	936
14.6	Handling Mouse Events	957
14.7	Timer Objects	967
14.8	Playing Audio	971
14.9	Common Errors to Avoid	976
	<i>Review Questions and Exercises</i>	976
	<i>Programming Challenges</i>	982

Chapter 15 **Recursion** 985

15.1	Introduction to Recursion	985
15.2	Solving Problems with Recursion	988
15.3	Examples of Recursive Methods	993
15.4	A Recursive Binary Search Method	999
15.5	The Towers of Hanoi	1002
15.6	Common Errors to Avoid	1007

<i>Review Questions and Exercises</i>	1007
<i>Programming Challenges</i>	1010

Chapter 16 Databases 1013

16.1	Introduction to Database Management Systems	1013
16.2	Tables, Rows, and Columns	1019
16.3	Introduction to the SQL <code>SELECT</code> Statement	1022
16.4	Inserting Rows	1044
16.5	Updating and Deleting Existing Rows.	1048
16.6	Creating and Deleting Tables	1057
16.7	Creating a New Database with JDBC	1060
16.8	Scrollable Result Sets	1062
16.9	Result Set Metadata	1063
16.10	Displaying Query Results in a <code>JTable</code>	1067
16.11	Relational Data.	1077
16.12	Advanced Topics	1099
16.13	Common Errors to Avoid	1101
<i>Review Questions and Exercises</i>		1101
<i>Programming Challenges</i>		1106

Java™ Quick Reference 1109

Index 1111

Credits 1127

Companion Website:

Appendix A	Getting Started with Alice
Appendix B	The ASCII/Unicode Characters
Appendix C	Operator Precedence and Associativity
Appendix D	Java Key Words
Appendix E	Installing the JDK and JDK Documentation
Appendix F	Using the <code>javadoc</code> Utility
Appendix G	More about the <code>Math</code> Class
Appendix H	Packages
Appendix I	Working with Records and Random Access Files
Appendix J	More about <code>JOptionPane</code> Dialog Boxes
Appendix K	Answers to Checkpoints
Appendix L	Answers to Odd-Numbered Review Questions
Appendix M	Installing Java DB
Case Study 1	Calculating Sales Commission
Case Study 2	The <code>Amortization</code> Class
Case Study 3	The <code>PinTester</code> Class
Case Study 4	Parallel Arrays
Case Study 5	The <code>FeetInches</code> Class
Case Study 6	The <code>SerialNumber</code> 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. 15 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. 106
Chapter 3	The <code>if</code> Statement, p. 109 The <code>if-else</code> Statement, p. 119 The <code>if-else-if</code> Statement, p. 130 The Time Calculator Problem, p. 189
Chapter 4	The <code>while</code> Loop, p. 197 The Pennies for Pay Problem, p. 267
Chapter 5	Passing Arguments to a Method, p. 283 Returning a Value from a Method, p. 297 The Retail Price Calculator Problem, p. 316
Chapter 6	Writing Classes and Creating Objects, p. 331 Initializing an Object with a Constructor, p. 352 The Personal Information Class Problem, p. 401
Chapter 7	Accessing Array Elements in a Loop, p. 411 Passing an Array to a Method, p. 426 The Charge Account Validation Problem, p. 491
Chapter 8	Returning Objects from Methods, p. 507 Aggregation, p. 519 The <code>BankAccount</code> Class Copy Constructor Problem, p. 556
Chapter 9	The Sentence Capitalizer Problem, p. 615
Chapter 10	Inheritance, p. 619 Polymorphism, p. 663 The <code>Employee</code> and <code>ProductionWorker</code> Classes Problem, p. 693
Chapter 11	Handling Exceptions, p. 699 The Exception Project Problem, p. 754
Chapter 12	Creating a Simple GUI Application, p. 758 Handling Events, p. 771 The Monthly Sales Tax Problem, p. 840

(continued on the next page)

LOCATION OF VIDEONOTES IN THE TEXT *(continued)*

Chapter 13	The <code>JList</code> Component, p. 846 The <code>JComboBox</code> Component, p. 862 The Image Viewer Problem, p. 906
Chapter 14	Creating an Applet, p. 923 The House Applet Problem, p. 982
Chapter 15	Reducing a Problem with Recursion, p. 989 The Recursive Power Problem, p. 1011

Preface

Welcome to *Starting Out with Java: From Control Structures through Objects*, Fifth 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 This 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:

- **New Java 7 Topics Added:** This edition discusses some of the new language features added in Java 7. Specifically, the ability to `switch` on a string expression is discussed in Chapter 3, type inference with the diamond operator is discussed in Chapter 7, and multi-catch exception handling is discussed in Chapter 11.
- **New Organization:** In previous editions, GUI programming with Swing was introduced in Chapter 7, and then revisited in Chapter 13. Many of the reviewers requested that GUI programming be postponed until inheritance had been covered. In this edition, the *First Look at GUI Applications* chapter has been moved to Chapter 12. However, the chapter has been written in a manner that it can still be covered immediately after Chapter 6, as in the previous editions. Instructors who prefer to follow the previous sequence of topics can still do so. (See the chapter dependency chart in Figure P-1 for possible sequencing of the chapters.)

- **A New Chapter on Databases:** New to this edition is Chapter 16, *Databases*. This chapter introduces the student to databases and SQL using JavaDB.
- **Expanded Coverage of the `switch` Statement:** In Chapter 3, the introduction to the `switch` statement has been rewritten, and enhanced with a flowchart. A section covering the ability to switch on a string in Java 7 has also been added.
- **Coverage of `System.out.printf` Has Been Expanded:** The section on `System.out.printf` in Chapter 3 has been completely rewritten, and expanded to include diagrams and coverage of additional format specifiers.
- **`System.out.printf` Is Used for Formatting Console Output:** In this edition, `System.out.printf` is used when numbers and strings need to be formatted for output in console programs. The `DecimalFormat` class is still introduced, but it is used to format numbers in GUI applications.
- **Discussion of Nested Loops Has Been Expanded:** In Chapter 4 the section on nested loops has been expanded to include an *In the Spotlight* section highlighting the use of nested loops to print patterns.
- **Usage of Random Numbers Has Been Expanded:** In Chapter 4 the section on random numbers has been expanded, and now includes *In the Spotlight* sections demonstrating how random numbers can be used to simulate the rolling of dice, and to determine the results of a coin toss.
- **A New Introduction to Objects Has Been Added to Chapter 6:** In Chapter 6, a new introduction to objects has been added to the beginning of the chapter. The new introduction is much more practical and concrete than the previous introduction, discussing Java objects that the student has already used. The goal of this new introduction is to show, in a familiar way, how programs consist of objects. This reinforces an object-oriented mind-set, and prepares the student to write his or her own classes.
- **New Motivational Examples of Classes Have Been Added to Chapter 6:** In Chapter 6, new motivational examples of classes have been added. One of the new examples introduces a `Die` class that simulates a die that can be rolled in a game. Another example shows how a variation of the game of Cho-Han can be simulated with classes that represent the players, a dealer, and the dice.
- **Equipping GUI Applications with a Static `main` Method Is Introduced Earlier:** In the *First Look at GUI Applications* chapter, which is now Chapter 12, the topic of equipping a GUI class with a static `main` method has been moved to a point very early in the chapter.
- **New Exercises and Programming Problems:** New shorter algorithm workbench exercises, and new motivational programming problems have been added to many of the chapters.

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.

Figure P-1 Chapter dependencies

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` methods. Formatting numeric output with the `DecimalFormat` class is covered, and the `System.out.printf` method is introduced.

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` and `StringTokenizer` classes.

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, and interfaces.

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: A First Look at GUI Applications. This chapter presents the basics of developing GUI applications with Swing. Fundamental Swing components and the basic concepts of event-driven programming are covered.

Chapter 13: Advanced GUI Applications. This chapter continues the study of GUI application development. More advanced components, menu systems, and look-and-feel are covered.

Chapter 14: Applets and More. In this chapter students apply their knowledge of GUI development to the creation of applets. In addition to using Swing applet classes, AWT classes are discussed for portability. Drawing simple graphical shapes is discussed.

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.pearsonhighered.com/gaddis.



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.pearsonhighered.com/gaddis. 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.pearsonhighered.com/gaddis.

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.pearsonhighered.com/gaddis:

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

Integrated Development Environment (IDE) Resource Kits

Professors who adopt this text for their students can also order an accompanying kit that contains the following popular Java development environments:

- Java™ SE Development Kit for Windows®
- Eclipse™ SDK for Windows®
- NetBeans™ IDE
- jGRASP™ IDE
- DrJava IDE
- BlueJ IDE
- TextPad® Text Editor for Windows®

The kit provides access to a Web site containing written and video tutorials for getting started in each IDE. For ordering information, please contact your campus Pearson Education representative or visit www.pearsonhighered.com/cs.

Online Practice and Assessment with MyProgrammingLab

MyProgrammingLab helps students fully grasp the logic, semantics, and syntax of programming. Through practice exercises and immediate, personalized feedback, MyProgrammingLab improves the programming competence of beginning students, who often struggle with the basic concepts and paradigms of popular high-level programming languages. A self-study and homework tool, the MyProgrammingLab course consists of hundreds of small practice problems organized around the structure of this textbook. For students, the system automatically detects errors in the logic and syntax of their code submissions and offers targeted hints that enable students to figure out what went wrong—and why. For instructors, a comprehensive gradebook tracks correct and incorrect answers and stores the code inputted by students for review.

MyProgrammingLab is offered to users of this book in partnership with Turing's Craft, the makers of the CodeLab interactive programming exercise system. For a full demonstration, to see feedback from instructors and students, or to get started using MyProgrammingLab in your course, visit www.myprogramminglab.com.

Instructor Resources

The following supplements are available 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.pearsonhighered.com/irc) or send an e-mail to computing@aw.com 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 of the Fourth Edition

Jesse Cecil
College of the Siskiyous

Kay Chen
Bucks County Community College

Jose Cordova
University of Louisiana, Monroe

David Goldschmidt
College of St. Rose

Nancy Harris
James Madison University

Alan Jackson
Oakland Community College (MI)

Laurie Murphy
Pacific Lutheran University

Elizabeth Riley
Macon State College

Yu Wu
University of North Texas

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

James Chegwidan
Tarrant County College

Kay Chen
Bucks County Community College

Brad Chilton
Tarleton State University

Diane Christie
University of Wisconsin, Stout

Cara Cocking
Marquette University

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

Georgia R. Grant
College of San Mateo

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

Norm Jacobson
University of California, Irvine

Stephen Judd
University of Pennsylvania

Harry Lichtbach
Evergreen Valley College

Michael A. Long
California State University, Chico

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

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

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

Peter van der Goes
Rose State College

Tuan A Vo
Mt. San Antonio College

Xiaoying Wang
University of Mississippi

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 Michael Hirsch and Matt Goldstein as my editors, and Chelsea Kharakozova 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 Yez Alayan as Marketing Manager, and Kathryn Ferranti as Marketing Coordinator. Their hard work is truly inspiring, and they do a great job getting my books out to the academic community. The production team of Marilyn Lloyd and Pat Brown 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, Alice, C++, Java™, Microsoft® Visual Basic®, Microsoft® Visual C#, and Python, all published by Pearson.

Introduction to Computers and Java

TOPICS

- | | |
|---|---------------------------------|
| 1.1 Introduction | 1.4 Programming Languages |
| 1.2 Why Program? | 1.5 What Is a Program Made Of? |
| 1.3 Computer Systems: Hardware and Software | 1.6 The Programming Process |
| | 1.7 Object-Oriented Programming |

1.1 Introduction

This book teaches programming using Java. Java is a powerful language that runs on practically every type of computer. It can be used to create large applications or small programs, known as applets, that are part of a Web site. Before plunging right into learning Java, however, this chapter will review the fundamentals of computer hardware and software, and then take a broad look at computer programming in general.

1.2 Why Program?

CONCEPT: Computers can do many different jobs because they are programmable.

Every profession has tools that make the job easier to do. Carpenters use hammers, saws, and measuring tapes. Mechanics use wrenches, screwdrivers, and ratchets. Electronics technicians use probes, scopes, and meters. Some tools are unique and can be categorized as belonging to a single profession. For example, surgeons have certain tools that are designed specifically for surgical operations. Those tools probably aren't used by anyone other than surgeons. There are some tools, however, that are used in several professions. Screwdrivers, for instance, are used by mechanics, carpenters, and many others.

The computer is a tool used by so many professions that it cannot be easily categorized. It can perform so many different jobs that it is perhaps the most versatile tool ever made. To the accountant, computers balance books, analyze profits and losses, and prepare tax reports. To the factory worker, computers control manufacturing machines and track production. To the mechanic, computers analyze the various systems in an automobile and pinpoint hard-to-find problems. The computer can do such a wide variety of tasks because it can

be *programmed*. It is a machine specifically designed to follow instructions. Because of the computer's programmability, it doesn't belong to any single profession. Computers are designed to do whatever job their programs, or *software*, tell them to do.

Computer programmers do a very important job. They create software that transforms computers into the specialized tools of many trades. Without programmers, the users of computers would have no software, and without software, computers would not be able to do anything.

Computer programming is both an art and a science. It is an art because every aspect of a program should be carefully designed. Here are a few of the things that must be designed for any real-world computer program:

- The logical flow of the instructions
- The mathematical procedures
- The layout of the programming statements
- The appearance of the screens
- The way information is presented to the user
- The program's "user friendliness"
- Manuals, help systems, and/or other forms of written documentation

There is also a science to programming. Because programs rarely work right the first time they are written, a lot of analyzing, experimenting, correcting, and redesigning is required. This demands patience and persistence of the programmer. Writing software demands discipline as well. Programmers must learn special languages such as Java because computers do not understand English or other human languages. Programming languages have strict rules that must be carefully followed.

Both the artistic and scientific nature of programming makes writing computer software like designing a car: Both cars and programs should be functional, efficient, powerful, easy to use, and pleasing to look at.

1.3

Computer Systems: Hardware and Software

CONCEPT: All computer systems consist of similar hardware devices and software components.

Hardware

Hardware refers to the physical components that a computer is made of. A computer, as we generally think of it, is not an individual device, but a system of devices. Like the instruments in a symphony orchestra, each device plays its own part. A typical computer system consists of the following major components:

- The central processing unit (CPU)
- Main memory
- Secondary storage devices
- Input devices
- Output devices

The organization of a computer system is shown in Figure 1-1.