

# 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:

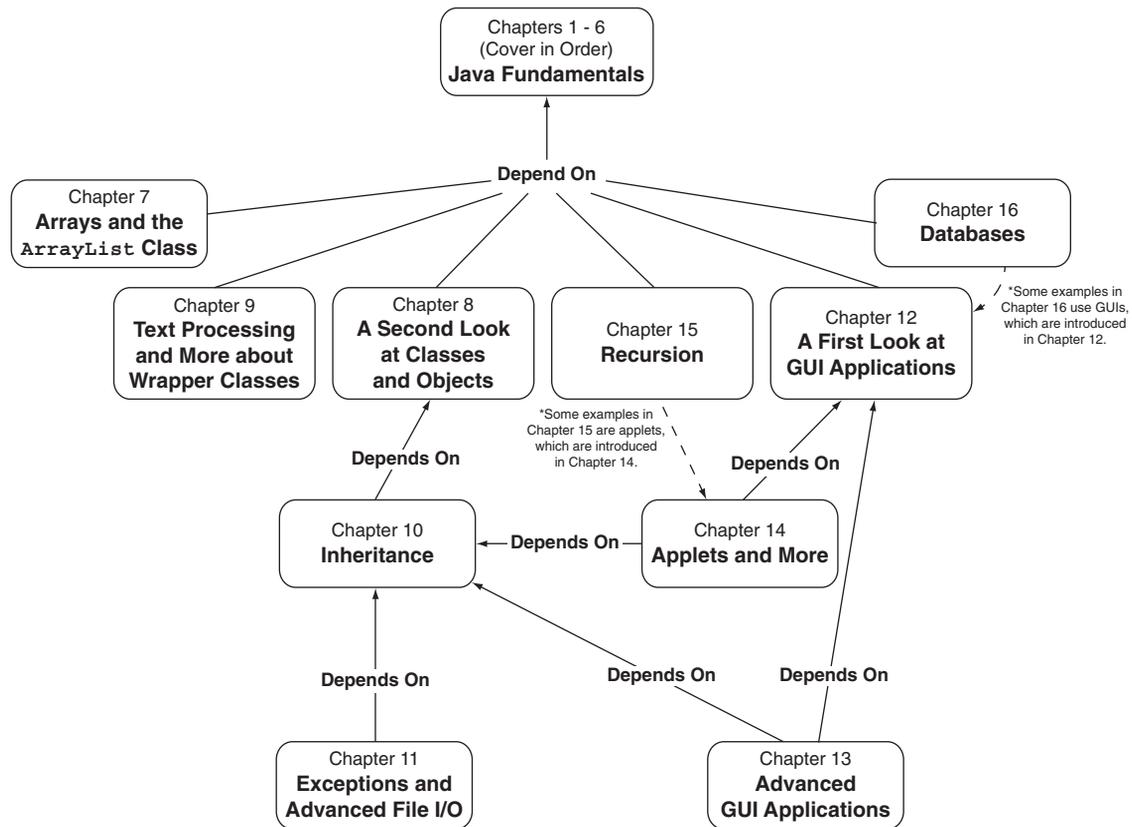
- **New Java 7 Topics Added:** This edition discusses some of the new language features added in Java 7. Specifically, the ability to `switch` on a string expression is discussed in Chapter 3, type inference with the diamond operator is discussed in Chapter 7, and multi-catch exception handling is discussed 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](http://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](http://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](http://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](http://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](http://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](http://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](http://www.pearsonhighered.com/irc)) or send an e-mail to [computing@aw.com](mailto: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 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*

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.