Preface

Welcome to *Starting Out with Java: From Control Structures through Data Structures,* Second Edition. This book is intended for a traditional two-semester CS1/CS2 sequence of courses. The first half of the book, intended for a CS1 class, teaches fundamental programming and problem-solving concepts using Java. The second half of the book, intended for a CS2 class, teaches advanced concepts and provides an introduction to algorithms and data structures. The book is written for students with no prior programming background, but experienced students will also benefit from its depth of detail.

Control Structures First, Then Objects, Then Data Structures

The text introduces students to the fundamentals of data types, input/output, control structures, methods, and objects created from standard library classes. First, students learn to write their own classes, and develop simple GUI applications. Next, students learn to use arrays. Then, a series of more advanced topics are presented, including inheritance, polymorphism, recursion, searching, sorting, algorithm analysis, and generics. Finally, students are introduced to the world of data structures, beginning with an overview of the Java Collections Framework (JCF), and continuing through a series of chapters in which students learn to develop their own data structures, including array-based lists, linked lists, stacks, queues, priority queues, binary trees, and AVL trees. Students are shown how to implement data structures either with or without generics, which gives instructors a great deal of flexibility in presenting the material.

From early in the book, applications are documented with `javadoc` comments. As students progress through the text, new `javadoc` tags are covered and demonstrated.

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

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:

- **Improved Organization in Chapter 3** The section covering the `if/else if` statement has been simplified and now appears immediately following the section on nested `if` statements. These sections have been rewritten to highlight the similarities between an `if/else if` statement and a nested `if` statement.

- **New *In the Spotlight* Sections** Many of the chapters have new sections titled *In the Spotlight*. Each of these provides a programming problem and a detailed, step-by-step analysis showing the student how to solve it.
- **Online VideoNotes** An extensive series of online videos has been developed to accompany this text. Throughout the book, VideoNote icons alert the student to videos covering specific topics. Additionally, one Programming Challenge at the end of each chapter now has an accompanying VideoNote explaining how to develop the problem's solution. The videos are available at `www.pearsonhighered.com/gaddis`.
- **Additional Programming Problems** Additional Programming Challenge problems have been added to many of the chapters. Several of these are simple games that will challenge and motivate students.

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. A solid-line arrow points from a chapter to the chapter that must be previously covered. A dotted-line arrow indicates that only a section or minor portion of the chapter depends on another chapter.

Brief Overview of Each Chapter

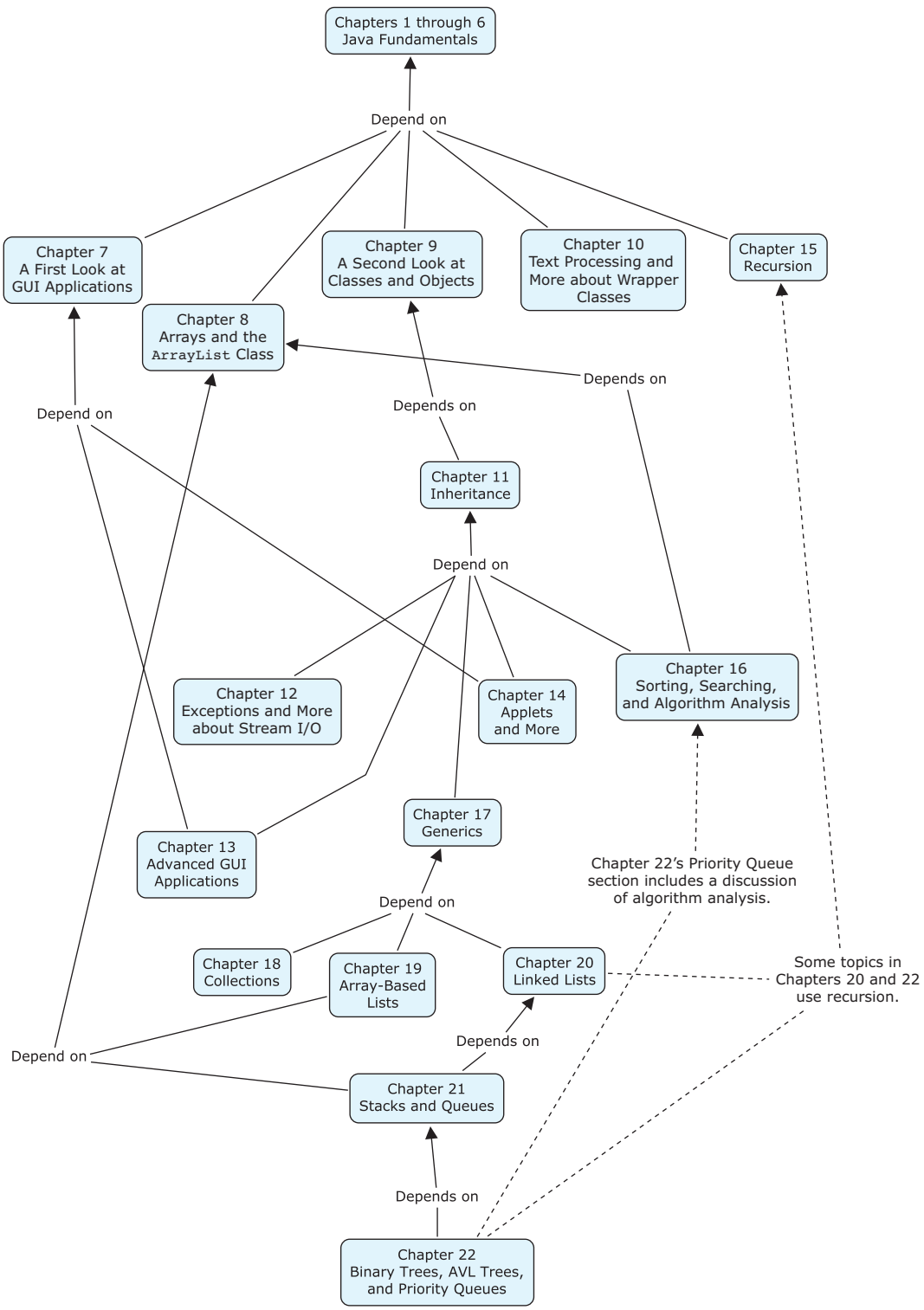
Chapter 1: Introduction to Computers and Java. This chapter provides an introduction to the field of computer science, and covers the fundamentals of hardware, software, and programming languages. The elements of a program, such as key words, variables, operators, and punctuation are discussed by examining 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.

Figure P-1 Chapter Dependencies



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: 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 8: 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 are also discussed. 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 9: 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 10: 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 `StringBuffer` and `StringTokenizer` classes.

Chapter 11: 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 12: Exceptions and More about Stream 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 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: Sorting, Searching, and Algorithm Analysis. In this chapter students learn the basics of sorting arrays and searching for data stored in them. The chapter covers the bubble sort, selection sort, insertion sort, Quicksort, sequential search, and binary search algorithms. A section on algorithm analysis is also provided, which covers basic steps, complexity, and Big O notation.

Chapter 17: Generics. This chapter shows students how to write generic classes and methods. Topics include erasure, passing instances of a generic class as arguments to a method, constraining type parameters, generics and inheritance, and generics and interfaces. Restrictions on the use of generic types are also discussed.

Chapter 18: Collections. This chapter introduces students to the Java Collections Framework (JCF). Lists, sets, maps, and the `Collections` class are discussed.

Chapter 19: Array-Based Lists. In this chapter students learn to write a resizable array-based list. First, students study a list that holds `String` objects, and then learn how generics can be used to create a list for holding objects of any type. Students also learn how to write an iterator for their own list classes.

Chapter 20: Linked Lists. This chapter introduces concepts and techniques for writing a linked list class. Basic linked list operations, such as adding, inserting, removing, and traversing, are covered. Doubly-linked lists, circularly-linked lists, and using recursion with linked lists are discussed.

Chapter 21: Stacks and Queues. In this chapter students learn about the operations of stacks and queues, and how to write array-based and linked list-based stack and queue classes.

Chapter 22: Binary Trees, AVL Trees, and Priority Queues. This chapter covers various aspects of binary trees and binary tree operations. The specific binary tree implementations covered are binary search trees, AVL trees, and priority queues.

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.

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 online videos, developed specifically for this book, are available for viewing 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 available for download from the Gaddis resource page at www.pearsonhighered.com/gaddis.

Common Errors to Avoid. Most chapters provide 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 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–K (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

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

Integrated Development Environment Resource Kits

Professors who adopt this text can order it for studetns with a kit containing seven popular Java IDEs (the most recent JDK from Oracle, Eclipse, NetBeans, jGRASP, DrJava, BlueJ, and TextPad) and access to a website 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.

myCodeMate—Your Own Teaching Assistant Is Just a Click Away

Pearson's *MyCodeMate* is a book-specific Web resource that provides tutorial help and evaluation of student programs. Selected Programming Challenges throughout the book have been integrated into *MyCodeMate* (indicated by the *MyCodeMate* icon). Using this tool, students are able to write and compile programs from any computer with Internet access, and receive guidance and feedback on how to proceed and on how to address compiler error messages. Instructors can track each student's progress on Programming Challenges from the text or can develop projects of their own. **A complimentary subscription to *MyCodeMate* is included when an access code is ordered packaged with a new copy of this text.** Subscriptions can also be purchased online. For more information about *MyCodeMate*, visit www.mycodemate.com, or contact your campus Pearson representative.

Visit the Pearson Instructor Resource Center (www.pearsonhighered.com/gaddis) or send an e-mail to ccomputing@pearson.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:

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. Daugherity
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

The authors would like to thank their families for their tremendous support during the development of this book. We also want to thank everyone at Pearson for making the *Starting Out With* series so successful. We are extremely fortunate to have Michael Hirsch as our editor and Stephanie Sellinger as editorial assistant. Michael's support and encouragement makes it a pleasure to write chapters and meet deadlines. We are also fortunate to have Yez Alayan as marketing manager and Kathryn Ferranti as marketing coordinator. They have done a great job getting this book out to the academic community. We had a great production team, led by Jeff Holcomb, managing editor, and Kayla Smith-Tarbox, production project manager. Thanks to you all!

About the Authors

Tony Gaddis is the principal author of the *Starting Out With* series of textbooks. Tony has nearly 20 years experience teaching computer science courses at Haywood Community College in North Carolina. He is a highly acclaimed instructor who was previously selected as the North Carolina Community College Teacher of the Year and has received the Teaching Excellence award from the National Institute for Staff and Organizational Development. The *Starting Out With* series includes introductory books using the C++ programming language, the Java™ programming language, Microsoft® Visual Basic®, Microsoft® C#®, Python, Programming Logic and Design, and Alice, all published by the Addison-Wesley imprint of Pearson.

Godfrey Muganda is an Associate Professor of Computer Science at North Central College. He teaches a wide variety of courses at the undergraduate and graduate levels including courses in Linux and Unix programming, Windows and .NET programming, web application development, web services, data structures, and algorithms. He is a past winner of the North Central College faculty award for outstanding scholarship. His primary research interests are in the area of fuzzy sets and systems.