Preface

This book is designed to serve as a textbook and reference for programming in the Java language. Although it does include programming techniques, it is organized around the features of the Java language rather than any particular curriculum of programming techniques. The main audience I had in mind when writing this book was undergraduate students who have not had extensive programming experience with the Java language. As such, it would be a suitable Java text or reference for either a first programming course or a later computer science course that uses Java. This book is designed to accommodate a wide range of users. The introductory chapters are written at a level that is accessible to beginners, while the boxed sections of those chapters serve to quickly introduce more experienced programmers to basic Java syntax. Later chapters are still designed to be accessible, but are written at a level suitable for students who have progressed to these more advanced topics.

CHANGES IN THIS EDITION

This fifth edition presents the same programming philosophy as the fourth edition. For instructors, you can teach the same course, presenting the same topics in the same order with no changes in the material covered or the chapters assigned. The changes to this edition consist almost exclusively of supplementary material added to the chapters of the previous edition, namely:

- Updates have been made for language changes in Java 7, such as allowing strings in switch statements.
- Twenty-five new programming projects have been added. By request, some of these are longer and less prescriptive projects to give the student more practice designing programming solutions.
- 15 new video notes have been created for a total of 46 video notes. These videos cover specific topics and offer solutions to the programming projects; they have been added to the book's website. The solutions walk students through the process of problem solving and coding to reinforce key programming concepts. An icon appears in the margin of the book when a video is available regarding the corresponding topic in the text.
- Chapter 2 now describes how to use the Scanner class to read from a text file so data-based programming projects can be explored prior to detailed coverage of File I/O in Chapter 10.
- A brief introduction to the Random class has been added to Chapter 3.
- Chapter 9 on exception handling begins with a new introduction of try/catch for handling input mismatch exceptions before discussing how to throw custom exceptions.
- A recursive algorithm to search the file system has been added to Chapter 11.
- Material on race conditions and thread synchronization has been added to Chapter 19.
- Ten new self-test exercises have been added along with the new material.

NO NONSTANDARD SOFTWARE

Only classes in the standard Java libraries are used. No nonstandard software is used anywhere in the book.

JAVA COVERAGE

All programs have been tested with Java 7. Oracle is not proposing any changes to future versions of Java that would affect the approach in this book.

OBJECT-ORIENTED PROGRAMMING

This book gives extensive coverage of encapsulation, inheritance, and polymorphism as realized in the Java language. The chapters on Swing GUIs provide coverage of and extensive practice with event driven programming. A chapter on UML and patterns gives additional coverage of OOP-related material.

FLEXIBILITY IN TOPIC ORDERING

This book allows instructors wide latitude in reordering the material. This is important if a book is to serve as a reference. It is also in keeping with my philosophy of writing books that accommodate themselves to an instructor's style rather than tying the instructor to an author's personal preference of topic ordering. With this in mind, each chapter has a prerequisite section at the beginning; this section explains what material must be covered before doing each section of the chapter. Starred sections, which are explained next, further add to flexibility.

STARRED SECTIONS

Each chapter has a number of starred (★) sections, which can be considered optional. These sections contain material that beginners might find difficult and that can be omitted or delayed without hurting the continuity of the text. It is hoped that eventually the reader would return and cover this material. For more advanced students, the starred sections should not be viewed as optional.

ACCESSIBLE TO STUDENTS

It is not enough for a book to present the right topics in the right order. It is not even enough for it to be clear and correct when read by an instructor or other expert. The material needs to be presented in a way that is accessible to the person who does not yet know the content. Like my other textbooks that have proven to be very popular, this book was written to be friendly and accessible to the student.

SUMMARY BOXES

Each major point is summarized in a short boxed section. These boxed sections are spread throughout each chapter. They serve as summaries of the material, as a quick reference source, and as a way to quickly learn the Java syntax for features the reader knows about in general but for which he or she needs to know the Java particulars.

SELF-TEST EXERCISES

Each chapter contains numerous Self-Test Exercises at strategic points in the chapter. Complete answers for all the Self-Test Exercises are given at the end of each chapter.

VIDEO NOTES

VideoNote

VideoNotes are step-by-step videos that guide readers through the solution to an end-of-chapter problem or further illuminate a concept presented in the text. Icons in the text indicate where a VideoNote enhances a topic. Fully navigable problems allow for self-paced instruction. VideoNotes are located at www.pearsonhighered.com/savitch.

OTHER FEATURES

Pitfall sections, programming tip sections, and examples of complete programs with sample I/O are given throughout each chapter. Each chapter ends with a summary section and a collection of programming projects suitable to assign to students.

ONLINE PRACTICE AND ASSESSMENT WITH MyProgrammingLab

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, a 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.

SUPPORT MATERIAL

The following support materials are available to all users of this book at www.pearsonhighered.com/cssupport:

- Source code from the book

The following resources are available to qualified instructors only at www.pearsonhighered.com/irc. Please contact your local sales representative for access information:

- Instructor's Manual with Solutions
- PowerPoint® slides

Integrated Development Environment Resource Kits

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

ACKNOWLEDGMENTS

Numerous individuals have contributed invaluable help and support in making this book happen: My former editor, Susan Hartman at Addison-Wesley, first conceived of the idea for this book and worked with me on the first editions; My current editor, Matt Goldstein, provided support and inspiration for getting subsequent editions reviewed, revised, and out the door; Chelsea Kharakozova, Marilyn Lloyd, Yez Alayan, and the other fine people at Pearson also provided valuable assistance and encouragement. Thanks also to GEX Publishing Services for their expert work in producing the final typeset book.

The following reviewers provided corrections and suggestions for this book. Their contributions were a great help. I thank them all. In alphabetical order they are:

Jim Adams	Chandler-Gilbert Community College
Gerald W. Adkins	Georgia College & State University
Dr. Bay Arinze	Drexel University
Prof. Richard G. Baldwin	Austin Community College
Kevin Bierre	Rochester Institute of Technology
Jon Bjornstad	Gavilan College
Janet Brown-Sederberg	Massasoit Community College
Tom Brown	Texas A&M University, Commerce
Charlotte Busch	Texas A&M University, Corpus Christi
Stephen Chandler	NW Shoals Community College
KY Daisy Fan	Cornell University
Adrienne Decker	University of Buffalo
Brian Downs	Century College
Keith Frikken	Miami University
Ahmad Ghafarian	North Georgia College & State University
Arthur Geis	College of DuPage
Massoud Ghyam	University of Southern California
Susan G. Glenn	Gordon College
Nigel Gwee	Louisiana State University
Judy Hankins	Middle Tennessee State University
May Hou	Norfolk State University

Sterling Hough	NHTI
Chris Howard	DeVry University
Eliot Jacobson	University of California, Santa Barbara
Balaji Janamanchi	Texas Tech University
Suresh Kalathur	Boston University
Edwin Kay	Lehigh University
Dr. Clifford R. Kettemborough	IT Consultant and Professor
Frank Levey	Manatee Community College
Xia Lin	Drexel University
Mark M. Meysenburg	Doane College
Sridhar P. Nerur	The University of Texas at Arlington
Hoang M. Nguyen	Deanza College
Rick Ord	University of California, San Diego
Prof. Bryson R. Payne	North Georgia College & State University
David Primeaux	Virginia Commonwealth University
Neil Rhodes	University of California, San Diego
W. Brent Seales	University of Kentucky
Lili Shashaani	Duquesne University
Riyaz Sikora	The University of Texas at Arlington
Jeff Six	University of Delaware
Donald J Smith	Community College of Allegheny County
Tom Smith	Skidmore College
Xueqing (Clare) Tang	Governors State University
Ronald F. Taylor	Wright State University
Thomas VanDrunen	Wheaton College
Shon Vick	University of Maryland, Baltimore County
Natalie S. Wear	University of South Florida
Dale Welch	University of West Florida
David A. Wheeler	
Wook-Sung Yoo	Gannon University

Special thanks goes to Kenrick Mock (University of Alaska Anchorage) who executed the updating of this edition. He once again had the difficult job of satisfying me, the editor, and himself. I thank him for a truly excellent job.

Walter Savitch