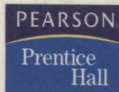


大学计算机教育国外著名教材系列 (影印版)



Java
AN INTRODUCTION TO COMPUTER
SCIENCE AND PROGRAMMING
Third Edition

**Java语言
计算机科学与程序设计
(第3版)**



Walter Savitch 著



清华大学出版社

TPS12
113

11/5/12
581

643125

大学计算机教育国外著名教材系列(影印版)

Java

An Introduction to Computer Science and Programming

Third Edition

Java 语言

计算机科学与程序设计

(第3版)

Walter Savitch

清华大学出版社
北京

English reprint edition copyright © 2004 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Java: An Introduction to Computer Science and Programming, Third Edition by Walter Savitch, Copyright © 2004

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall, Inc.

This edition is authorized for sale and distribution only in the People's Republic of China(excluding the Special Administrative Region of Hong Kong, Macao SAR and Taiwan).

本书影印版由 Pearson Education(培生教育出版集团)授权给清华大学出版社出版发行。

For sale and distribution in the People's Republic of China exclusively(except Taiwan, Hong Kong SAR and Macao SAR).

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

北京市版权局著作权合同登记号 图字 01-2003-8787

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Java 语言:计算机科学与程序设计(第3版)=Java: An Introduction to Computer Science and Programming, Third Edition/(美)萨维特(Savitch, W.)著. —影印本. —北京:清华大学出版社,2004.4

(大学计算机教育国外著名教材系列)

ISBN 7-302-08303-7

I. J… II. 萨… III. Java 语言—程序设计—高等学校—教材—英文 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 020939 号

出版者:清华大学出版社

地址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮编:100084

社总机:(010)62770175

客户服务:(010)62776969

责任编辑:龙啟铭

印刷者:清华大学印刷厂

装订者:三河市金元装订厂

发行者:新华书店总店北京发行所

开本:185×230 印张:60.75

版次:2004年4月第1版 2004年4月第1次印刷

书号:ISBN 7-302-08303-7/TP·5983

印数:1~5000

定价:86.00元(含光盘)

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

出版说明

进入 21 世纪,世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才,谁就能在竞争中取得优势。高等教育,作为培养高素质人才的事业,必然受到高度重视。目前我国高等教育的教材更新较慢,为了加快教材的更新频率,教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从 1996 年开始,与国外著名出版公司合作,影印出版了“大学计算机教育丛书(影印版)”等一系列引进图书,受到了国内读者的欢迎和支持。跨入 21 世纪,我们本着为我国高等教育教材建设服务的初衷,在已有的基础上,进一步扩大选题内容,改变图书开本尺寸,一如既往地请有关专家挑选适用于我国高校本科及研究生计算机教育的国外经典教材或著名教材,组成本套“大学计算机教育国外著名教材系列(影印版)”,以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材,以利我们把“大学计算机教育国外著名教材系列(影印版)”做得更好,更适合高校师生的需要。

清华大学出版社

FEATURES OF THIS TEXT

FAQ: Why Do We Need new?

When `new` is used in an expression such as the following, you can think of it as creating the instance variables of the object.

```
SpeciesFirstTry speciesOfLastMonth = new SpeciesFirstTry();
```

An object of a class type, such as `speciesOfLastMonth`, can have smaller variables inside of it, namely, the instance variables of the object. The `new` places these instance variables inside of the object. We will explain this use of `new` more completely in Section 4.3.

Gotcha ▶

Helps students identify possible programming pitfalls.

▲ Gotcha

Variables Declared in a Block

When you declare a variable within a block, that variable becomes a local variable for the block. This means that you cannot use the variable outside of the block. If you want to use a variable outside of a block, you must declare it outside of the block. Declaring the variable outside of the block will let you use the variable both outside and inside the block. △

◀ FAQ

Provides students answers to frequently asked questions within the context of the chapter.

Quick Reference: Method Invocation (Calling a Method)

You *invoke* a method by writing the calling object followed by a dot, then the name of the method, and finally a set of parentheses that may (or may not) have information to pass to the method.

◀ Quick Reference

Provides concise, user-friendly reference for key programming concepts.

Java Tip ▶

Gives students helpful information about the Java programming language.

■ Java Tip

Use of return in void Methods

A `void` method returns no value and so is not required to have any `return` statement. However, there is a kind of `return` statement that you may sometimes want to use in a `void` method. A `return` statement within a `void` method has the form

```
return;
```

Remember: Two Kinds of Methods

There are two kinds of methods: (1) those that return a single value and (2) those that perform some action other than return a value. Methods that perform some action other than returning a value are called `void methods`.

◀ Remember

Reinforces concepts presented in the chapter.

Self-Test Questions ▶

Provides students the opportunity to practice skills learned in the chapter.

? Self-Test Questions

1. Consider the program in Display 4.4. Suppose you wanted to add another species object called `speciesOfTheYear`, and suppose you wanted the user to give it data, specifically a name, population, and growth rate. What code do you need to add to the program? (*Hint*: It requires only three or four lines of code.)
2. Suppose `Employee` is a class with a `void` method named `readInput` and `dilbert` is an object of the class `Employee`. So `dilbert` was named and created by the following:

```
Employee dilbert = new Employee();
```

PREFACE FOR INSTRUCTORS

This book was designed to be used in a first course in programming and computer science. It covers programming techniques, as well as the basics of the Java programming language. It is suitable for courses as short as one quarter or as long as one full academic year. No previous programming experience is required, nor is any mathematics, other than a little high school algebra. The book can also be used for a course designed to teach Java to students who have already had another programming course, in which case the first few chapters can be assigned as outside reading. (For students who have had previous programming experience in C or C++, there is an appendix that explains some of the differences between Java and C or C++.)

All of the code in the book has been tested with Java 2 from Sun Microsystems (version 1.4). To be fully compatible with the material presented, the Java used in your class should be version 1.4 or higher. The coverage of Java was carefully arrived at by class testing and is a concise, accessible introduction for beginners.

Changes in This Edition

If you have not used the second edition of this text, you can skip this subsection. If you have used the second edition, this subsection will tell you how this third edition differs from that edition.

For instructors, the transition from the second edition of the text to this third edition is easy: You can teach the same course, with basically the same topics presented, in the same order and with only very minor changes in the material covered.

In addition to technical changes in the material, which we discuss shortly, we have completely gone over and refined the presentation for this edition. Without abandoning our student-friendly, easy-to-understand style, we have redone the entire text to make it more concise. Some have complained about the quality of the copyediting in the previous edition. We have gone to great pains to obtain a copy editor we have used before and know to be an excellent editor. Here are the main differences between this and the previous edition:

We have added coverage of the new assertion-checking facility now built into Java. This topic is covered in Chapter 4.

Binary file I/O is now covered using the classes `ObjectInputStream` and `ObjectOutputStream`, rather than the classes `DataInputStream` and `DataOutputStream`. For I/O of primitive types and strings, the details are the same in either case. For I/O of class objects, `ObjectInputStream` and `ObjectOutputStream` are more object oriented and, even more important, are easier to use and perform better.

The previous edition allowed for covering either text files first or binary files first. This required some repetition of material. Many instructors disliked the repetition, and as far as we can now tell, nobody was choosing the option of covering binary files first. This edition requires that text files be covered before binary files. As a result, the discussion is more concise and is organized in the way that instructors prefer. (You can, of course, skip coverage of binary files completely if you wish.)

This edition adds coverage of the Unified Modeling Language (UML). All UML coverage is optional (even though it is not labeled as such).

The material on `JOptionPane` that was in Chapter 2 of the second edition has been moved to Appendix 10 in this edition. Instructors told us that they did not cover the material, so we removed it from Chapter 2 to simplify that chapter. For any instructors who do want to teach the material, we left it in an appendix that can be covered anytime after Chapter 2.

Chapter 15, which covered more advanced graphics techniques, such as drawing pictures, advanced color manipulation, and control of fonts, has been eliminated. Instructors told us that they never used the material and that they would prefer a shorter book. Chapters 12, 13, and 14 still include extensive coverage of Swing windowing interfaces and applets.

The CD in the book now comes with Java 2 SDK edition, version 1.4, for Windows and Linux and with Sun ONE Studio 4, Community Edition Integrated Development Environment IDE (formerly Forte for Java 4, Community Edition), for Windows and Linux. The CD also contains a simpler shareware IDE named TextPad, which runs under Windows and which beginners may find more friendly and easier to learn. Because we have added Sun ONE Studio 4 (aka Forte), there was no longer any need for JBuilder, so we have omitted the trial version of JBuilder from the CD for this edition. JBuilder is a fine IDE and will work well with the text, but we felt that a third IDE choice might be too confusing for beginning students. If you prefer JBuilder, you can have your students download a trial version from Borland's Web site (<http://www.borland.com/> as we went to press).

Latest Java Coverage

This edition has been updated to use the latest features of Java, including assertion checking and the updated versions of the Swing and file I/O classes. As we wrote the book, we checked all code on Sun's Java 2 version 1.4.

Flexible Sequence of Topics

If you are an instructor, this book adapts the material presented to the way you teach, rather than making you adapt to the material. The book does not tightly prescribe the order in which your course must cover topics, nor does it prescribe the specialized libraries that must be used in your course. You can easily change the order in which you cover chapters and sections. The details about rearranging material are explained in a dependency chart following the acknowledgements section. More details are given in a "Prerequisites" section at the start of each chapter.

Since Java does not include any simple console input, most texts (even those which are more advanced) provide a class for console input. This book adds only one simple class for console input, thus requiring as little nonstandard software as possible. Even that one class, which is included early in the book, becomes an understandable programming example for students well before the end of the book. All of the remaining software is from standard Java libraries that should be part of any Java installation.

Coverage of Problem-Solving and Programming Techniques

The book is designed to teach students basic problem-solving and programming techniques and is not simply a book about Java syntax. Numerous case studies and programming tips, as well as many other sections, explain important problem-solving and programming techniques, such as loop design techniques, debugging techniques, style techniques, abstract data types, and basic object-oriented programming techniques, including UML and event-driven programming, as well as other computer science topics.

Object-Oriented and Traditional Techniques

Any course that really teaches Java must teach classes early, since everything in Java involves classes. A Java program is a class. The data type for strings of characters is a class. Even the behavior of the equals operator (==) depends on whether it is comparing objects from classes or simpler data items. Classes cannot be avoided, except by means of absurdly long and complicated “magic formulas.” Accordingly, the book introduces classes fairly early. Some exposure to using classes is given in Chapters 1 and 2. Chapter 4 covers how to define classes. All of the basic information about classes—including inheritance—is presented by the end of Chapter 7 (even if you omit Chapter 6). However, some topics regarding classes (including inheritance) can be postponed to later in a course.

Although the book introduces classes early, it does not neglect traditional programming techniques, such as top-down design and loop design techniques. These older topics may no longer be glamorous, but they are information that all beginning students need.

UML Coverage

New to this edition is the coverage of UML, which starts in Chapter 4. Instructors who would prefer to leave UML coverage to a later course can skip the material without losing continuity of the text.

Swing GUIs and Applets

Starting with the first version of Java 2, Java comes with an improved graphical user interface (GUI) library known as Swing that allows programmers to design portable GUIs. This book uses Swing to teach students to produce professional-looking windowing interfaces. In the process, students learn event-driven programming, as well as receiving a lot of practice with object-oriented programming.

As we class-tested this material and gathered the views of instructors, we found that regular Swing GUIs were a more accessible way than applets to teach students object-oriented programming. Thus, we place greater emphasis on regular Swing GUIs in this text. Such an emphasis makes sense, since almost all advanced applet tools are really general Swing tools. However, for those who want to cover applets early, Chapter 1 has an optional section that previews them. Chapter 13 covers applets in detail and may be presented much earlier than the chapter number suggests. You may choose to introduce GUIs early, late, or not at all.

In addition to the optional GUI material in Chapter 1, there are three full chapters on GUIs, thus giving thorough coverage of Swing GUIs and applets.

Language Details and Sample Code

The book teaches programming technique, rather than simply the Java language. However, neither students nor instructors would be satisfied with an introductory programming course that did not also teach the programming language. Until you calm a student's fears about language details, it is often impossible to focus her or his attention on bigger issues. For this reason, the book gives complete explanations of Java language features and lots of sample code. Programs are presented in their entirety, along with sample input and output. In many cases, there are even extra complete examples on the CD, in addition to the complete examples in the text.

Self-Test Questions

Self-test questions are spread throughout each chapter. These questions have a wide range of difficulty levels. Some require only a one-word answer, whereas others require the reader to write an entire nontrivial program. Complete answers to all the self-test questions, including those requiring full programs, are given at the end of each chapter.

Class Tested

The material in the book has been fully class tested. Much of the material and many of the methods of presentation were revised in response to this testing.

Support Material

The support materials described here that are not included with the book can be obtained from the publisher or over the Internet.

Companion CD

Each book contains a CD that includes all the programs and classes in the volume. The CD also includes Sun Microsystems' Java 2 SDK edition, version 1.4, for Windows and Linux and Sun ONE Studio 4, Community Edition (formerly Forte for Java 4, Community Edition), for Windows and Linux. The CD also includes a copy of TextPad, a simpler shareware IDE that runs under Windows and that may be easier for students to deal with than the Sun ONE Studio 4 IDE (previously known as Forte).

Instructor's Resource Guide and Companion Web Site

The instructor's tools include a chapter-by-chapter *Instructor's Resource Guide* that contains numerous teaching hints, quiz questions with solutions, and solutions to many programming exercises. The companion Web site includes code, PowerPoint slides, and other

teaching resources. Instructors should contact their Prentice Hall sales representative to obtain a copy of the *Instructor's Resource Guide* and to receive information on how to access the companion Website. For the name and number of your sales representative, call Prentice Hall Faculty Services at 1-800-526-0485. Additional information on this book and other Prentice Hall products can be found on Prentice Hall's Web site at

<http://www.prenhall.com/>

Other Resources

There are many Java resources available from other vendors which we do not supply, but which you may wish to purchase. One such resource is the JJ environment from Public Static Void Main. This environment includes a simple IDE as well as course administration software all accessed via the Internet. The JJ environment has been setup to be highly compatible with this book. In particular, the SavitchIn class, which is used for keyboard input in this book, is an installed library class in JJ. For more information see their website:

<http://www.LearnJavaNow.org/>

Walter Savitch

wsavitch@ucsd.edu

<http://www.cse.ucsd.edu/users/savitch>

PREFACE FOR STUDENTS

This book is designed to teach you the Java programming language and, even more importantly, to teach you basic programming techniques. It requires no previous programming experience and no mathematics other than some simple high school algebra. However, to get the full benefit of the book, you should have a version of Java available on your computer, so that you can practice with the examples and techniques given. You should have a version of Java called Java 2. That “2” is not a version number, but something more like a generation number. The version number will be of the form 1.n.x, such as 1.2.x or 1.4.x. The “x” is another number and may not even be there. To be fully compatible with the book, your Java version number should be 1.4.x or higher. (The exact number that is filled in for the “x” is not critical. Indeed, the “x” need not even be present. If software says only “version 1.4,” that is fine.)

If You Have Programmed Before

You need not have any previous programming experience to use this book. It was designed for beginners. Nonetheless, you can still use the book to learn Java if you happen to have had experience with some other programming language—but allow us to give you a few words of advice. Do not assume that Java is the same as the programming language(s) you are accustomed to using. All languages are different, and the differences, even if small, are large enough to give you problems. Read at least the boxed material in Section 1.3 of Chapter 1 and all of the boxed material in Chapters 2 and 3. These boxed sections are labeled “Quick Reference,” “Remember,” and “FAQ.” By the time you reach Chapter 4, you would be wise to read the entire chapter.

If you have programmed before in either C or C++, the transition to Java can be troublesome. At first glance, Java may seem almost the same as C or C++. However, Java is very different from these languages, and you need to be aware of the differences. Appendix 11 presents a comparison of Java and C++ that will help you see what the differences are.

Copies of the Programs in the Text

Accompanying the book is a CD that includes all the programs and other software examples given in the volume, so that you can practice with these examples without having to type them into your computer.

Obtaining a Copy of Java

The version of Java you use depends somewhat on what operating system you are using. Be sure to consult the subsection which follows that corresponds to your operating system.

Microsoft Windows

The CD that comes with this text contains a copy of the Java 2 SDK edition, version 1.4, and the Sun ONE Studio 4 (also known as Forte). The Java 2 SDK includes the Java compiler. The Sun ONE Studio 4 (also called Forte) is an IDE that includes an editor from which you can compile and run Java programs. Install both pieces of software, and you will have all you need to write and run Java programs.

The CD also contains a trial version of the TextPad IDE as an alternative to the Sun ONE Studio 4 (also called Forte). Both IDEs are good in their own way, and you may, of course, use either one, but our preference is for the TextPad IDE, which is simpler to use and has all you need to work with the material in the text. If you are in a course, take your instructor's advice on what IDE you should use. If the choice is yours, we suggest that you use the TextPad IDE. Note that in order to use TextPad, you must still install the Java software from Sun described in the previous paragraph.

Macintosh Operating Systems

A recent version of Java for the Mac OS X can be downloaded from the Apple website. When this book went to press, the URL was:

<http://developer.apple.com/java/>

As we went to press this Mac Java was Java 1.3.1, which should be fine for almost all the code in this book. When the version 1.4.1 Java for the Mac comes out, we suggest that you change to it.

Another good alternative is to purchase a version of CodeWarrior from Metrowerks, Inc. It works well with the Mac operating system.

Linux Operating System

The CD that comes with this text contains a copy of the Java 2 SDK edition, version 1.4, and the Sun ONE Studio 4 (also known as Forte). The Java 2 SDK includes the Java compiler. The Sun ONE Studio 4 (also called Forte) is an IDE that includes an editor from which you can compile and run Java programs. Install these pieces of software, and you will have all you need to write and run Java programs. (Note that, if you prefer, you may use an editor other than the Sun ONE Studio 4 (Forte).)

UNIX Operating Systems (Other than Linux)

If you are in a course using UNIX, then, in all likelihood, Java has already been installed on the course computer for you. If Java has not been installed, visit the Sun Web site to download a suitable version of Java. At the time this book went to press, the URL was

<http://java.sun.com/>

Self-Test Questions

Each chapter contains numerous self-test questions. Complete answers to all the questions are given at the end of the chapter. One of the best ways to practice what you are learning is to do the self-test questions *without looking at the answers*. Look at the answers only after you have answered the questions.

This Text Is Also a Reference Book

In addition to using this book as a textbook, you can and should use it as a reference. When you need to check a particular point that you may have forgotten or that you hear mentioned by somebody, but have not yet learned yourself, just look in the index. Many index entries give a page number for "quick reference." Turn to this quick-reference page. It will contain a short entry, usually set off in a box, that gives all the essential points on that topic. You can consult the quick-reference page to check details of the Java language, as well as details on programming techniques.

Boxed sections in every chapter give you a quick summary of the main points in that chapter. These boxed sections are labeled “Quick Reference,” “Remember,” and “FAQ.” You can use these boxes to review the chapter, preview the chapter, or check details of the Java language.

We Want Your Opinions

This book was written for you, and I would like to hear any comments you have on it. You can contact me via e-mail at the following address:

wsavitch@ucsd.edu

Unfortunately, I cannot provide you with answers to the programming exercises. Only instructors who adopt the book can receive (selected) answers from the publisher. For help with the programming exercises, you will have to contact your instructor. (Even if you are not enrolled in a class, I still cannot provide answers to programming exercises.) But remember that there are answers to all the self-test questions at the end of each chapter.

Walter Savitch

<http://www.cse.ucsd.edu/users/savitch>

Acknowledgments

I thank the Computer Science and Engineering Department of the University of California, San Diego (UCSD), which is my home department and the place that I tested much of the material covered in this book. Many students in my classes were kind enough to help correct preliminary versions of the book. These students’ comments and the comments of instructors who class-tested the book were a tremendous help in shaping the final text. In particular, I extend a special thanks to Carole McNamee of California State University, Sacramento, and to Paul Kube of UCSD; their feedback and class-testing of earlier editions or drafts of the book were a great help to me in producing this edition.

I also thank all the reviewers who took the time to read drafts of this or the previous editions of the book. They provided invaluable detailed comments and suggestions. In alphabetical order within each group, they are as follows:

Reviewers for this third edition:

Robert P. Burton—Brigham Young University
 Steve Cater—Kettering University
 Gobi Gopinath—Suffolk County Community College.
 Rob Kelly—SUNY, Stony Brook
 Michele Kleckner—Elon College
 Mike Litman—Western Illinois University
 Michael Olan—Stockton State
 Ken Slonneger—University of Iowa
 Boyd Trolinger—Butte College

In addition, the following individuals were kind enough to provide helpful responses to a survey done in preparation for this third edition:

Martin Chelten—Moorpark Community College
 Tom Cortina—SUNY, Stony Brook
 Adel Elmaghraby—University of Louisville

Gopal Gupta—University of Texas, Dallas
 Ricci Heishman—North Virginia Community College
 Blayne Mayfield—Oklahoma State University
 Subramanian Vijayarangam—University of Massachusetts, Lowell

Reviewers for the second edition:

Jim Buffenbarger—Idaho State University
 Martin Chetlen—Moorpark Community College
 Tom Cortina—SUNY, Stony Brook
 Prasun Dewan—University of North Carolina
 Laird Dornan—Sun Microsystems, Inc.
 H. E. Dunsmore—Purdue University, Lafayette
 Adel Elmaghraby—University of Louisville
 Gopal Gupta—New Mexico State University
 Le Gruenwald—University of Oklahoma
 Ric Heishman—North Virginia Community College
 Rob Kelly—SUNY, Stony Brook
 Blayne Mayfield—Oklahoma State University
 Alan Saleski—Loyola University, Chicago

Reviewers for the first edition:

Michael Clancy—University of California, Berkeley
 Michael Godfrey—Cornell University
 Robert Herrmann—Sun Microsystems, Inc., Java Soft
 Robert Holloway—University of Wisconsin, Madison
 Lily Hou—Carnegie Mellon University
 John Motil—California State University, Northridge
 James Roberts—Carnegie Mellon University
 Nan C. Schaller—Rochester Institute of Technology
 Ryan Shoemaker—Sun Microsystems, Inc.
 Donald E. Smith—Rutgers University

I also thank all the individuals at Prentice Hall who organized the reviewing and production of this book. In particular, I thank Toni Holm for her work in coordinating things between offices and I thank both Lakshmi Balasubramanian and Xiaohong Zhu for their fine job of converting my manuscript into a published book. Thanks also to Patrick Lindner, Heather Scott, and Jake Warde. All these wonderful people cheerfully did a great job. I extend a special thanks to my publisher, Alan Apt, for his invaluable support and advice throughout the writing and production process.

Lew Rakocy prepared the programming solutions for the instructor's guide, Brian Durney prepared the PowerPoint slides and the test bank questions. I thank both of them for their conscientious work.

I thank Rebecca Pepper for an excellent job of copyediting the manuscript. Since I had final approval on all corrections, any problems that remain in the text are my fault.

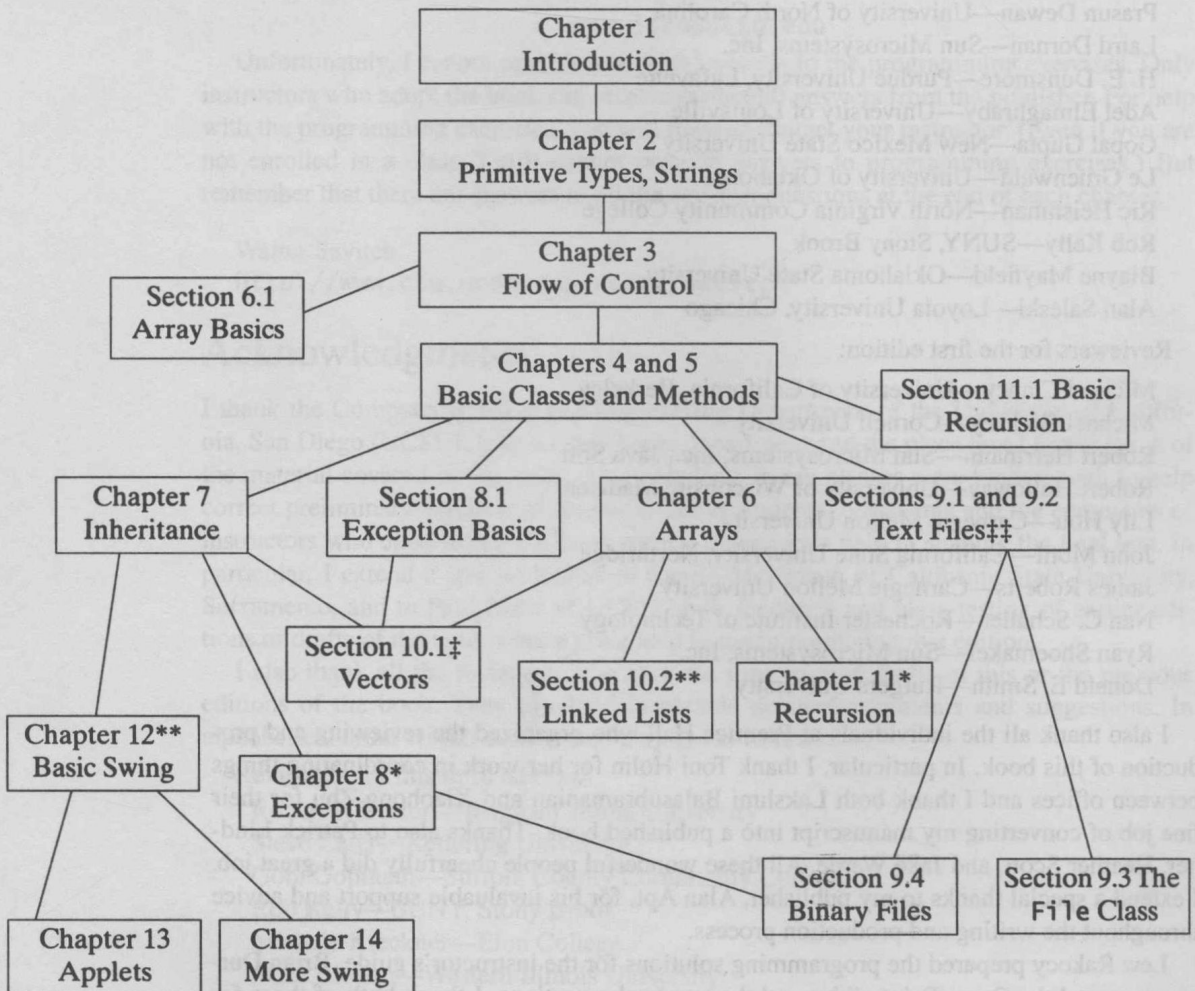
I thank Sun Microsystems for allowing me to use the Duke icon in a number of my GUI examples and for permitting us to include Java software on the CD.

Finally, I give a special thanks to Christina for her help and inspiration.

W.S.

Dependency Chart

This chart shows the prerequisites for the chapters in the book. If there is a line between two boxes, the material in the higher box should be covered before the material in the lower box. Minor variations to the chart are discussed in the “Prerequisites” section at the start of each chapter. These variations usually provide more, rather than less, flexibility than what is shown on the chart.



* Note that some sections of these chapters can be covered sooner. Those sections are given in this chart.

** See the chapter’s “Prerequisites” section for full details.

‡ Most of Section 10.1 (“Vectors”) can be covered before covering inheritance.

‡‡ Also requires Section 8.1 of Chapter 8.

BRIEF TABLE OF CONTENTS

Chapter 1	Introduction to Computers and Java 3
Chapter 2	Primitive Types, Strings, and Console I/O 39
Chapter 3	Flow of Control 103
Chapter 4	Defining Classes and Methods 181
Chapter 5	More About Objects and Methods 261
Chapter 6	Arrays 341
Chapter 7	Inheritance 415
Chapter 8	Exception Handling 475
Chapter 9	Streams and File I/O 541
Chapter 10	Dynamic Data Structures 611
Chapter 11	Recursion 669
Chapter 12	Window Interfaces Using Swing 709
Chapter 13	Applets and HTML 795
Chapter 14	More Swing 823
Appendix 1	Keywords 892
Appendix 2	Precedence Rules 893
Appendix 3	Unicode Character Set 894
Appendix 4	SavitchIn 895
Appendix 5	Protected and Package Modifiers 908
Appendix 6	The DecimalFormat Class 909
Appendix 7	The Iterator Interface 912
Appendix 8	Cloning 913
Appendix 9	Javadoc 916
Appendix 10	The JOptionPane Class 918
Appendix 11	Differences between C++ and Java 924
Index	927

TABLE OF CONTENTS



CHAPTER 1 Introduction to Computers and Java 3

Objectives 4

Prerequisites 4

1.1 Computer Basics 4

Hardware and Memory 5

Programs 7

Programming Languages and Compilers 8

Java Byte-Code 9

Linking 12

1.2 Designing Programs 12

Object-Oriented Programming 13

Encapsulation 14

Polymorphism 15

Inheritance 16

If You Know Some Other Programming Language 17

Algorithms 17

Reusable Components 18

Testing and Debugging 18

Gotcha: Coping with “Gotchas” 19

Gotcha: Hidden Errors 20

1.3 A Sip of Java 20

History of the Java Language 20

Applets 21

A First Java Application Program 22

Compiling a Java Program or Class 28

Running a Java Program 29

Preview Examples of Applets (*Optional*) 30

Chapter Summary 33

Answers to Self-Test Questions 34

Programming Projects 36

CHAPTER 2 Primitive Types, Strings, and Console I/O 39

Objectives 40

Prerequisites 40

2.1 Primitive Types and Expressions 40

Variables 41

Java Identifiers 44

Gotcha: Java Is Case-Sensitive 46

Primitive Types 46

Assignment Statements 47

Specialized Assignment Operators 49

Simple Input and Output 49

Number Constants 51

Assignment Compatibilities 52

Type Casting 53

