Introduction to

Java™

Programming

with JBuilder 4

Y. Daniel Liang
Department of Computer Science
Armstrong Atlantic State University

An Alan R. Apt Book

Prentice Hall
Upper Saddle River, New Jersey 07458
http://www.prenhall.com
To Samantha, Michael, and Michelle
To the Instructor

Java Teaching Strategies

There are three popular strategies in teaching Java. The first is to mix Java applets and graphics programming with object-oriented programming concepts. The second is to introduce object-oriented programming from the start. The third strategy is a step-by-step approach, first laying a sound foundation on programming concepts, control statements, and methods, then introducing object-oriented programming, and finally moving on to graphical user interface (GUI), applets, internationalization, multithreading, multimedia, I/O, and networking.

The first strategy, starting with GUI and applets, seems attractive, but requires substantial knowledge of object-oriented programming and a good understanding of the Java event-handling model; thus, students may never fully understand what they are doing. The second strategy is based on the notion that objects should be introduced first because Java is an object-oriented programming language. This notion, however, overlooks the importance of the fundamental techniques required for writing programs in any programming language. From my own experience, confirmed by the experiences of many colleagues, I have found that learning basic logic and fundamental programming techniques like loops is a struggle for most freshmen. Students who cannot write code in procedural programming are not able to learn object-oriented programming. A good introduction on primitive data types, control statements, methods, and arrays prepares students to learn object-oriented programming. Therefore, this text adopts the third strategy, first proceeding at a steady pace through all the necessary and important basic concepts, then moving to object-oriented programming, and then to using the object-oriented approach to build interesting GUI applications and applets with exception handling, internationalization, multithreading, multimedia, I/O, networking, and data structures.

Selection of Java Subjects

Many introductory Java texts lack sufficient breadth and do not cover all the core Java knowledge that is needed to develop useful projects. Some authors are over ambitiously mix too many topics, such as Java database programming, Remote Method Invocation, JavaBeans and Rapid Application Development, servlets, and JSP, into one introductory Java text. With this approach the coverage of programming principles tends to lose coherence. What is the basis for deciding that one approach is too light and the other too heavy? I believe that the best yardstick is the Level 1 Java Certification Exam initiated by a consortium of leading IT companies, including Sun Microsystems, IBM, Oracle, Hewlett-Packard, BEA Systems, and Sybase. The Level 1 Certification Exam tests core Java knowledge and fundamental programming skill.

This book gives a comprehensive introduction on the fundamentals of programming in Chapters 1–5, an in-depth treatment of object-oriented programming in
Chapters 6–9, extensive examples of graphics programming in Chapters 10–12, and appropriate coverage of advanced Java topics in Chapters 13–19. The book covers all the subjects required for the Level 1 Java Certification Exam.

**Audience of This Book**

The book is suited for both beginning and advanced students, depending on how it is used. It has been used in two-semester freshman programming courses and one-semester courses in Java as a second language. It has also been used in short training courses for experienced programmers. Computer science departments, engineering departments, and management information systems departments around the world have used this book at various levels. For students with no programming experience, an entire semester of four credit hours could be spent just on the first five chapters of the book, as we do for the first programming course at Armstrong Atlantic State University.

**Instructor Resources**

The Instructor’s Manual on CD-ROM is available for instructors using this book. It contains the following resources:

- Improved Microsoft PowerPoint slides for lectures, with interactive buttons to view syntax-highlighted source code and to run programs without leaving the slides.
- Abundant exercises. All of the solutions are provided in the Instructor’s Manual. Students will have access to the solutions of even-numbered exercises in the book’s companion CDROM.
- Twelve sample exams. In general, each exam has four parts: (1) multiple-choice questions or short questions; (2) correct programming errors; (3) trace programs; (4) write programs.
- Suggested syllabi for teaching Java to freshman, for teaching Java as a second language, and for teaching Java to corporate employees.
- Lecture notes. A number of suggested teaching strategies and activities are presented to help you in the delivery of the course.

To obtain the Instructor’s Manual, contact your Prentice-Hall sales representative.

Microsoft PowerPoint slides, answers to review questions, solutions to even-numbered programming exercises, and source code for the examples in the book are available at the book’s companion Web site at [www.cs.armstrong.edu/liang/introjb4.html](http://www.cs.armstrong.edu/liang/introjb4.html) and [www.prenhall.com/liang](http://www.prenhall.com/liang). The Web site also contains interactive online self-tests and other supplemental materials.

**Pedagogical Features of the Book**

Introduction to Java Programming with JBuilder 4 uses the following elements to get the most out of the material:
Objectives lists what students should have learned from the chapter. This will help them to determine whether they have met the objectives after completing the chapter.

Introduction opens the discussion with a brief overview of what to expect from the chapter.

Programming concepts are taught by representative Examples, carefully chosen and presented in an easy-to-follow style. Each example is described, and includes the source code, a sample run, and an example review. The source code of the examples is contained in the companion CD-ROM.

Each program is complete and ready to be compiled and executed. The sample run of the program is captured from the screen to give students a live presentation of the example. Reading these examples is much like entering and running them on a computer.

Chapter Summary reviews the important subjects that students should understand and remember. It helps them to reinforce the key concepts they have learned in the chapter.

Review Questions help students to track their progress and evaluate their learning.

Programming Exercises at the end of each chapter provide students with opportunities to apply the skills on their own. The trick of learning programming is practice, practice, and practice. To that end, the book provides a large number of exercises.

Notes, Tips, and Cautions are inserted throughout the text to offer valuable advice and insight on important aspects of program development.

**NOTE**
Provides additional information on the subject and reinforces important concepts.

**Tip**
Teaches good programming style and practice.

**CAUTION**
Helps students steer away from the pitfalls of programming errors.

**What's New in This Edition**
This book expands and improves upon *Introduction to Java Programming with JBuilder 3*. The major changes are as follows:
The book is updated to JBuilder 4.

UML graphical notations are used to describe classes and their relationships throughout the book. UML has become a standard methodology for class design. The use of UML enhances the treatment of object-oriented design and programming.

Part II, "Objected-Oriented Programming," is expanded into four chapters to provide an in-depth introduction to object-oriented programming and design. Strings are now in a separate chapter, and Chapter 9, "Object-Oriented Software Development," introduces class design.

Every chapter has been thoroughly revised and improved. Much of Chapter 9, "Object-Oriented Software Development," has been newly written to introduce object-oriented development using the UML approach and foster the concept of developing reusable components. Chapter 19, "Java Data Structures," introduces the Java collections framework, is brand-new.

The book is reorganized to provide flexible ordering of chapters. Arrays are covered in Chapter 5, but can be taught after Chapter 7. Input and Output are covered in Chapter 17, but they can be taught right after Chapter 13, "Exception Handling." Java Data Structures are explained in Chapter 19, but can be taught after Chapter 9, "Object-Oriented Software Development." See the Chapter Dependency Chart for other possible sequences.

Several new appendices provide readers with additional background information and supplemental material.

The comprehensive companion Web site includes an interactive self-test for each chapter at www.cs.armstrong.edu/liang/introjb4.html. There are over 500 questions in the self-test.

To the Student

There is nothing more important to the future of computing than the Internet. There is nothing more exciting on the Internet than Java. A revolutionary programming language developed by Sun Microsystems, Java has become the de facto standard for cross-platform applications and programming on the World Wide Web.

Before Java, the Web was used primarily for viewing static information on the Internet using HTML, a mark-up language for document layout and for linking documents over the Internet. Java programs can be embedded in an HTML page and downloaded by Web browsers to bring live animation and interactive applications to Web clients. Java can also be used on the server-side to generate dynamic HTML content to the clients.

Java is a full-featured, general-purpose programming language that is capable of developing robust mission-critical applications. In the last three years, Java has gained enormous popularity and has quickly become the most popular and successful pro-
gramming language. Today, it is used not only for Web programming, but also for developing standalone applications. Many companies that once considered Java to be more hype than substance are now using it to create distributed applications accessed by customers and partners across the Internet. For every new project being developed today, companies are asking how they can use Java to make their work easier.

Java’s Design and Advantages

Java is an object-oriented programming language. Object-oriented programming is a favored programming approach that has replaced traditional procedure-based programming techniques. An object-oriented language uses abstraction, encapsulation, inheritance, and polymorphism to provide great flexibility, modularity, and reusability for developing software.

Java is platform-independent. Its programs can run on any platform with a Java virtual machine, a software component that interprets Java instructions and carries out associated actions.

Java is distributed. Networking is inherently built-in. Simultaneous processing can occur on multiple computers on the Internet. Writing network programs is treated as simple data input and output.

Java is multithreaded. Multithreading is the capability of a program to perform several tasks simultaneously, such as downloading a video file while playing the video at the same time. Multithreading is particularly useful in graphical user interfaces (GUI) and network programming. Multithread programming is smoothly integrated in Java. In other languages, you can only enable multithreading by calling procedures that are specific to the operating system.

Java is secure. Computers become vulnerable when they are connected with other computers. Viruses and malicious programs can damage your computer. Java is designed with multiple layers of security that ensure proper access to private data and restrict access to disk files.

Java’s Versatility

Stimulated by the promise of writing programs once and running them anywhere, Java has become the most ubiquitous programming language. Java programs run on full-featured computers, and also on consumer electronics and appliances such as Palm and mobile phones.

Because of its great potential to unite existing legacy applications written on different platforms so that they can run together, Java is perceived as a universal frontend for the enterprise database. The leading database companies, IBM, Oracle, and Sybase, have extended their commitment to Java by integrating it into their products. Oracle, for example, enables Java applications to run on its server, and to deliver a complete set of Java-based development tools supporting the integration of current applications with the Web.
Learning Java

To first-time programmers, learning Java is like learning any high-level programming language. The fundamental point in learning programming is to develop the critical skills of formulating programmatic solutions for real problems and translating the solutions into programs using selection statements, loops, and methods.

Applying the concept of abstraction in the design and implementation of software projects is the key to developing software. The overriding objective of this book, therefore, is to teach students to use many levels of abstraction in solving problems and to see problems in small and in large. The examples and exercises throughout the book foster the concept of developing reusable components and using them to create projects.

Students with no programming experience should take a slow-paced approach in Part I of the book. I recommend that you complete all the exercises in Part I before moving to Chapter 6. Students new to object-oriented programming may need some time to become familiar with the concept of objects and classes. Once the principles are mastered, programming in Java is easy and productive. Students who know object-oriented programming languages like C++ and Smalltalk will find it easier to learn Java. In fact, Java is simpler than C++ and Smalltalk in many aspects.

Learning Java with JBuilder

You can use Java 2 SDK to write Java programs. Java 2 SDK (formerly known as JDK) consists of a set of separate programs, such as compiler and interpreter, each of which is invoked from a command line. Besides Java 2 SDK, there are more than a dozen Java development tools on the market today, including Borland JBuilder, Sun Forte, IBM Visual Age for Java, Microsoft Visual J++, and WebGain Visual Café. These tools support an integrated development environment (IDE) for rapidly developing Java programs. Editing, compiling, building, debugging, and online help are integrated in one graphical user interface. Using these tools effectively will greatly increase your programming productivity.

The overriding objective of this book is to introduce the concepts and practice of Java programming. To facilitate developing and managing Java programs, the book is aided by JBuilder. With a tool like JBuilder, students can not only develop Java programs more productively, but can also learn Java programming more effectively.

JBuilder is a premier Java development tool for developing Java programs produced by Borland. Borland products are known to be “best of breed” in the IDE tool market. Over the years, it has led the charge in creating visual development tools like Delphi and C++ Builder. Borland is now leading the way in Java development tools with JBuilder. JBuilder is endorsed by major information technology companies like Oracle, which licensed JBuilder 2.
JBuilder is easy to learn and easy to use. The JBuilder development team worked hard to simplify the user interface and make it easy to navigate through the programs, projects, classes, packages, and code elements. As a result, JBuilder has fewer windows than other Java IDE tools. This makes JBuilder an ideal tool for beginners and for students who have little programming experience.

JBuilder is an indispensable, powerful tool that boosts your programming productivity. It may take a while to become familiar with it, but the time you invest will pay off in the long run. This text takes an incremental approach to facilitate learning JBuilder. Programming with JBuilder is introduced throughout the book to help you gradually adapt to using it.

**Note**
This book uses JBuilder 4. You can also use JBuilder 5, since their user interfaces are almost identical.

**Organization of the Book**

This book is divided into four parts that, taken together, form a comprehensive introductory course on Java programming. Because knowledge is cumulative, the early chapters provide the conceptual basis for understanding Java and guide students through simple examples and exercises; subsequent chapters progressively present Java programming in detail, culminating with the development of comprehensive Java applications. The appendixes contain a mixed bag of topics, including an HTML tutorial.

**Part I: Fundamentals of Programming**

The first part of the book is a stepping stone that will prepare you to embark on the journey of learning Java. You will begin to know Java, and will learn how to write simple Java programs with primitive data types, control statements, methods, and arrays.

**Chapter 1**, "Introduction to Java and JBuilder 4," gives an overview of the major features of Java: object-oriented programming, platform-independence, Java bytecode, security, performance, multithreading, and networking. The chapter also introduces JBuilder and uses it to create, compile, and run Java programs.

**Chapter 2**, "Primitive Data Types and Operations," introduces primitive data types, operators, and expressions. Important topics include identifiers, variables, constants, assignment statements, assignment expressions, primitive data types, operators, and shortcut operators. Java programming style and documentation are also addressed. You will learn how to run Java programs from the command line, and how to get online help from JBuilder.

**Chapter 3**, "Control Statements," introduces decision and repetition statements. Java decision statements include various forms of if statements, and the switch
statement. Java repetition statements include the while loop, the do loop, and the for loop. The keywords break and continue are discussed. You will learn how to customize JBuilder IDE options.

Chapter 4, "Methods," introduces method creation, calling methods, passing parameters, returning values, method overloading, scope of local variables, and recursion. Applying the concept of abstraction is the key to developing software. The chapter also introduces the use of method abstraction in problem-solving. The Math class for performing basic math operations is introduced. Various JBuilder commands in the Search, View, Build, and Run menus are discussed.

Chapter 5, "Arrays," explores an important structure: arrays for processing data in lists and tables. You will learn how to declare, initialize, and copy arrays. Examples of using two-dimensional arrays for matrix operations are provided. This chapter also introduces popular search and sorting methods. You will also learn how to use the JBuilder debugger.

Part II: Object-Oriented Programming

In the book’s second part, object-oriented programming is introduced. Java is a class-centric, object-oriented programming language that uses abstraction, encapsulation, inheritance, and polymorphism to provide great flexibility, modularity, and reusability in developing software. You will learn programming with objects and classes, class inheritance, interfaces, polymorphism, and developing software using the object-oriented approach.

Chapter 6, "Objects and Classes," begins with objects and classes. The important topics include defining classes, creating objects, using constructors, passing objects to methods, instance and class variables, and instance and class methods, scope of variables in the context of a class, the keyword this, and using the UML graphical notations to describe classes. Several examples are provided to demonstrate the power of the object-oriented programming approach. Students will learn the benefits (abstraction, encapsulation, and modularity) of object-oriented programming from these examples.

Chapter 7, "Strings," introduces the classes String, StringBuffer, and StringTokenizer for storing and processing strings. There are more than 1500 predefined Java classes grouped in several packages. Starting with this chapter, students will gradually learn how to use Java classes to develop their own programs. The classes on strings are fine examples to demonstrate the concept of objects and classes.

Chapter 8, "Class Inheritance and Interfaces," teaches how an existing class can be extended and modified as needed. Inheritance is an extremely powerful programming technique, further extending software reusability. Java programs are all built by extending predefined Java classes. The major topics include defining subclasses, using the keyword super, using the modifiers protected, final, and abstract, polymorphism and dynamic binding, and casting objects. This chapter introduces the object class, which is the root of all Java classes. You will also learn how to use abstract classes and interfaces.
Chapter 9, "Object-Oriented Software Development," focuses on class design. You will learn how to analyze relationships among objects, and design classes with the relationships association, aggregation, composition, strong inheritance, and weak inheritance. This chapter gives the guidelines for class design with several examples. The wrapper classes for primitive data types are introduced to encapsulate primitive data type values as objects. Finally, two examples of designing generic classes for matrix operations and linked lists are introduced.

Part III: Graphics Programming

The third part of the book introduces Java graphics programming. Major topics include event-driven programming, creating graphical user interfaces, and writing applets. You will learn the architecture of Java graphics programming API and use the user interface components to develop graphics applications and applets.

Chapter 10, "Getting Started with Graphics Programming," introduces the concepts of Java graphics programming using Swing components. Topics include the Swing class hierarchy, frames, panels, and simple layout managers (FlowLayout, GridLayout, and BorderLayout). The chapter introduces drawing geometric figures in the graphics context. The concept and techniques for Java event-driven programming are presented. You will also learn how to use the JBuilder Application wizard to create Java applications.

Chapter 11, "Creating User Interfaces," introduces the user interface components: buttons, labels, text fields, text areas, combo boxes, lists, check boxes, radio buttons, menus, scrollbars, and scroll panes. Today's client/server and Web-based applications use a graphical user interface. Java has a rich set of classes to help you build GUIs.

Chapter 12, "Applets and Advanced Graphics," takes an in-depth look at applets, discussing applet behavior and the relationship between applets and other Swing classes. Applets are a special kind of Java class that can be executed from the Web browser. Students will learn how to run programs both as applications and as applets. The chapter also introduces two advanced layout mangers (CardLayout and GridBagLayout) and the use of no layout. Advanced examples of handling mouse and keyboard events are provided. You will also learn to create applets using the JBuilder Applet wizard.

Part IV: Developing Comprehensive Projects

The book's final part is devoted to several advanced features of Java programming. You will learn how to use these features to develop comprehensive programs; for example, using exception handling to make your program robust, using multi-threading to make your program more responsive and interactive, incorporating sound and images to make your program user-friendly, using input and output to manage and process a large quantity of data, creating client/server applications with Java networking support, and using the classes in the Java Collections Framework to build data structures in Java.
Chapter 13, "Exception Handling," teaches students how to define exceptions, throw exceptions, and handle exceptions so that their programs can either continue to run or terminate gracefully in the event of runtime errors. The chapter discusses predefined exception classes, and gives examples of creating user-defined exception classes.

Chapter 14, "Internationalization," introduces the development of Java programs for international audiences. You will learn how to format dates, numbers, currencies, and percentages for different regions, countries, and languages. You will also learn how to use resource bundles to define which images and strings are used by a component depending on the user’s locale and preferences.

Chapter 15, "Multithreading," introduces threads, which enable the running of multiple tasks simultaneously in one program. Students will learn how to use the Thread class and the Runnable interface to launch separate threads. The chapter also discusses thread states, thread priority, thread groups, and the synchronization of conflicting threads.

Chapter 16, "Multimedia," teaches how to incorporate sound and images to bring live animation to Java applets and applications. Various techniques for smoothing animation are introduced.

Chapter 17, "Input and Output," introduces input and output streams. Students will learn the class structures of I/O streams, byte and character streams, file I/O streams, data I/O streams, print streams, object streams, random file access, delimited I/O, and interactive I/O.

Chapter 18, "Networking," introduces network programming. Students will learn the concept of network communication, stream sockets, client/server programming, and reading data files from the Web server.

Chapter 19, "Java Data Structures," introduces the Java Collections Framework. Students will learn to use classes and interfaces such as Collection, Set, HashSet, TreeSet, Iterator, List, ArrayList, LinkedList, Vector, Stack, Map, HashMap, TreeMap, Collections, and Arrays to build projects.

Appendixes

This part of the book covers a mixed bag of topics. Appendix A lists Java keywords. Appendix B gives tables of ASCII characters and their associated codes in decimal and in hex. Appendix C shows the operator precedence. Appendix D summarizes Java modifiers and their usage. Appendix E introduces number systems and conversions among binary, decimal, and hex numbers. The bitwise operations are also introduced in this appendix. Appendix F introduces HTML basics. Appendix G lists UML Graphical Notations for describing classes and their relationships. Appendix H introduces JavaBeans and rapid Java application development using JBuilder. Appendix I discusses special floating-point values. Finally, Appendix J provides a glossary of key terms used in the text.
Chapter Dependency Chart

The new edition provides flexible ordering of chapters. You may cover Chapter 6, "Objects and Classes," after Chapter 4, "Methods." You may cover Chapter 17, "Input and Output," after exception handling is introduced in Chapter 13. Chapter 19, "Java Data Structures," can be covered after Chapter 9, "Object-Oriented Software Development."
ABOUT THE AUTHOR

Y. Daniel Liang is the author of the Prentice-Hall Liang Java Series. He has taught more than fifty Java courses to university students and corporate employees.

Dr. Liang is currently a Yamacraw professor of software engineering in the Department of Computer Science at Armstrong Atlantic State University, Savannah, Georgia. He can be reached via the Internet at liangjava@yahoo.com or liang@armstrong.edu.

ACKNOWLEDGMENTS

I am grateful to the readers of my Java Series, both students and instructors, who offered comments, suggestions, bug reports, and praise. Their enthusiastic support contributed to the success of these books.

This edition was greatly improved thanks to outstanding reviews by Professors Roger Kraft of Purdue University at Calumet, Kevin Parker of Idaho State University, Lixin Tao of Concordia University, and David Champion and Hong Lin, both at the DeVry Institute.

My students and colleagues at Armstrong Atlantic State University have been great supporters of this ongoing project. I would like to thank Ray Greenlaw of Armstrong Atlantic State University for his support of incorporating Java into the CS curriculum, and for his tireless efforts to build a first-class undergraduate computer science program.

This edition has benefited from the previous editions of all my Java books. I would like to acknowledge the following people who contributed to the previous editions: Bill Morrison, Hao Wu, Greg Geller, Ben Page, Michael Willig, Russell Minnich, Balaram Nair, Ben Stonebraker, C-Y Tang, Bertrand I-P Lin, Maw-Shang Chang, Ruay-Shiung Chang, Mike Sunderman, Fen English, James Silver, Mark Temte, Bob Sanders, Marta Partington, Tom Cirtin, Songlin Qiu, Tim Tate, Carolyn Linn, Alfonso Hermida, Nathan Clement, Eric Miller, Chris Barrick, John Etchison, Louisa Klucznik, Angela Denny, Randy Haubner, Robin Drake, Betsy Brown, and Susan Kindel.

For this edition, I would like to thank Alan Apt, Toni Holm, Amy Todd, Jennie Burger, Cindy Szollose, Jake Warde, and their colleagues at Prentice-Hall for organizing and managing this project, and Patty Donovan, Robert Milch, Dan Boilard, Dana Smith, and their colleagues at Pine Tree Composition for helping to produce the book.

As always, I am indebted to my wife, Samantha, for love, support, and encouragement.
COMPANION WEB SITE FOR THE BOOK

The companion Web site for the book can be accessed from www.cs.armstrong.edu/liang/introjb4.html or www.prenhall.com/liang. The Web site contains the following resources:

- Interactive Self-Test for every chapter
- Answers to review questions
- Solutions to even-numbered programming exercises
- Source code for the examples in the book
- Java Programming Style and Documentation Guidelines
- JBuilder Supplements (Implement Interface wizard, Override Method wizard, Archive Builder, and Resource Strings)
- Microsoft PowerPoint slides for lectures
- Errata
- FAQs

STUDENT CD-ROM

The student CD-ROM comes with the book. The contents of the CD-ROM are the following:

- JBuilder 4 Foundation from Borland. To activate JBuilder, you need to obtain a serial number and a key from www.borland.com/jbuilder/foundation/download/.
- Answers to review questions
- Solutions to even-numbered programming exercises
- Source code for the examples in the book
JBUILDER 4 TOPICS COVERED IN THE BOOK

All JBuilder 4 features covered in the book are available in JBuilder 4 Foundation. My Web site covers additional features available in JBuilder 4 Professional, such as Implement Interface wizard, Override Method wizard, Archive Builder, and Resource Strings.

Chapter 1, "Introduction to Java and JBuilder 4"
  Introducing the JBuilder user interface
  Creating a project in JBuilder
  Creating a program, compiling, and running a program

Chapter 2, "Primitive Data Types and Operations"
  Running Java programs from the command line
  Getting online help from JBuilder

Chapter 3, "Control Statements"
  Customizing JBuilder IDE options

Chapter 4, "Methods"
  JBuilder menu commands in the Search, View, Build, and Run

Chapter 5, "Arrays"
  Debugging in JBuilder 4

Chapter 10, "Getting Started with Graphics Programming"
  Debugging GUI programs
  Using the Application wizard

Chapter 12, "Applets and Advanced Graphics"
  Using the Applet wizard

Appendix H, "JavaBeans and Rapid Java Application Development Using JBuilder 4"
  Using the UI designer
  Using the menu designer
# Contents at a Glance

## Part I  Fundamentals of Programming  1
- Chapter 1  Introduction to Java and JBuilder  3
- Chapter 2  Primitive Data Types and Operations  39
- Chapter 3  Control Statements  83
- Chapter 4  Methods  129
- Chapter 5  Arrays  175

## Part II  Object-Oriented Programming  215
- Chapter 6  Objects and Classes  217
- Chapter 7  Strings  265
- Chapter 8  Class Inheritance and Interfaces  287
- Chapter 9  Object-Oriented Software Development  337

## Part III  Graphics Programming  381
- Chapter 10  Getting Started with Graphics Programming  383
- Chapter 11  Creating User Interfaces  453
- Chapter 12  Applets and Advanced Graphics  531

## Part IV  Developing Comprehensive Projects  593
- Chapter 13  Exception Handling  595
- Chapter 14  Internationalization  625
- Chapter 15  Multithreading  663
- Chapter 16  Multimedia  699
- Chapter 17  Input and Output  729
- Chapter 18  Networking  783
- Chapter 19  Java Data Structures  833

## Appendices  863
- Appendix A  Java Keywords  865
- Appendix B  The ASCII Character Set  867
- Appendix C  Operator Precedence Chart  871
- Appendix D  Java Modifiers  875
- Appendix E  Number Systems and Bit Manipulations  877
- Appendix F  An HTML Tutorial  883
- Appendix G  UML Graphical Notations  899
- Appendix H  JavaBeans and Rapid Java Application Development Using JBuilder  903
- Appendix I  Special Floating-Point Values  921
- Appendix J  Glossary  923

## Index  931
# Table of Contents

## Part I  Fundamentals of Programming  

### Chapter 1  Introduction to Java and JBuilder  

1. Introduction  
2. The History of Java  
3. Characteristics of Java  
   - Java Is Simple  
   - Java Is Object-Oriented  
   - Java Is Distributed  
   - Java Is Interpreted  
   - Java Is Robust  
   - Java Is Secure  
   - Java Is Architecture-Neutral  
   - Java Is Portable  
   - Java’s Performance  
   - Java Is Multithreaded  
   - Java Is Dynamic  
3. Java and the World Wide Web  
4. The Java Language Specification  
5. Java Development Tools and JBuilder  
6. Getting Started with JBuilder  
   1. The Main Menu  
   2. The Toolbar  
   3. The Status Bar  
   4. The Project Pane  
   5. The Content Pane  
   6. The Structure Pane  
7. Creating a Project  
8. A Simple Java Program  
   1. Creating a Java Program  
   2. Compiling a Java Program  
   3. Executing a Java Application  
9. Anatomy of a Java Program  
   1. Comments  
   2. Packages  
   3. Reserved Words  

---

xxi
CHAPTER 2 Primitive Data Types and Operations

Introduction 40
Writing Simple Programs 40
Identifiers 45
Variables 46
Declaring Variables 46
Assignment Statements and Assignment Expressions 47
Declaring and Initializing Variables in One Step 48
Constants 48
Numeric Data Types 49
Numeric Operators 49
Numeric Literals 50
Arithmetic Expressions 51
Shortcut Operators 51
Numeric Type Conversions 53
Character Data Type 55
Casting between char and Numeric Types 56
boolean Data Type 57
Operator Precedence and Associativity 60
Operand Evaluation Order 61
Case Studies 62
Programming Style and Documentation 66
Appropriate Comments and Comment Styles 66
Naming Conventions 67
Proper Indentation and Spacing 67
Block Styles 68
CHAPTER 3 Control Statements

Introduction 84
Selection Statements 84
  Simple if Statements 84
  if . . . else Statements 86
  Nested if Statements 87
  switch Statements 90
  Conditional Expressions 92
Loop Statements 93
  The while Loop 93
  The do Loop 96
  The for Loop 97
Using the Keywords break and continue 104
  Statement Labels and Breaking with Labels 108
Case Studies 108
Customizing JBuilder Environment 113

CHAPTER 4 Methods

Introduction 130
Creating a Method 130
Calling a Method 131
Passing Parameters 134
  Pass by Value 134
Overloading Methods 137
  Ambiguous Invocation 139
The Scope of Local Variables 139
Method Abstraction 141
The Math Class 141
  Trigonometric Methods 141
  Exponent Methods 142
  The Rounding Methods 142
  The min, max, abs, and random Methods 143
### Arrays

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>176</td>
</tr>
<tr>
<td>Declaring Array Variables and Creating Arrays</td>
<td>176</td>
</tr>
<tr>
<td>Initializing and Processing Arrays</td>
<td>178</td>
</tr>
<tr>
<td>Passing Arrays to Methods</td>
<td>182</td>
</tr>
<tr>
<td>Copying Arrays</td>
<td>187</td>
</tr>
<tr>
<td>Multidimensional Arrays</td>
<td>190</td>
</tr>
<tr>
<td>Sorting Arrays</td>
<td>193</td>
</tr>
<tr>
<td>Searching Arrays</td>
<td>196</td>
</tr>
<tr>
<td>The Linear Search Approach</td>
<td>196</td>
</tr>
<tr>
<td>The Binary Search Approach (Optional)</td>
<td>198</td>
</tr>
<tr>
<td>Debugging in JBuilder (Optional)</td>
<td>201</td>
</tr>
<tr>
<td>Starting the Debugger</td>
<td>203</td>
</tr>
<tr>
<td>Controlling Program Execution</td>
<td>204</td>
</tr>
<tr>
<td>Examining and Modifying Data Values</td>
<td>206</td>
</tr>
<tr>
<td>Setting Breakpoints</td>
<td>208</td>
</tr>
<tr>
<td>Debugging Selection SortWithError</td>
<td>209</td>
</tr>
</tbody>
</table>

### PART II  OBJECT-ORIENTED PROGRAMMING

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects and Classes</td>
<td>217</td>
</tr>
<tr>
<td>Introduction</td>
<td>218</td>
</tr>
<tr>
<td>Defining Classes for Objects</td>
<td>218</td>
</tr>
<tr>
<td>Creating Objects and Object Reference Variables</td>
<td>220</td>
</tr>
<tr>
<td>Differences between Variables of Primitive Types</td>
<td>221</td>
</tr>
<tr>
<td>and Reference Types</td>
<td>221</td>
</tr>
<tr>
<td>Garbage Collection</td>
<td>222</td>
</tr>
</tbody>
</table>
## Table of Contents

### CHAPTER 7  Strings  265

- Introduction
- The `String` Class
  - Constructing a String
  - String Comparisons
  - String Concatenation
  - Extracting Substrings
  - String Length and Retrieving Individual Characters
  - String Conversions
  - Converting Characters and Numeric Values to Strings
- The `StringBuffer` Class
  - Constructing a String Buffer
  - Modifying Strings in the Buffer
  - The `capacity`, `length`, `setLength`, and `charAt` Methods
- The `StringTokenizer` Class
- Command-Line Arguments
  - Passing Arguments to Java Programs
  - Processing Command-Line Parameters

### CHAPTER 8  Class Inheritance and Interfaces  287

- Introduction
- Superclasses and Subclasses
- Using the `super` Keyword
  - Calling Superclass Constructors
  - Calling Superclass Methods
- Overriding Methods
- The `Object` class
Creating a Frame 387
Centering a Frame (Optional) 389
Adding Components to a Frame 390
Layout Managers 391
FlowLayout 391
GridLayout 394
BorderLayout 396
Using Panels as Containers 398
Drawing Graphics in Panels 400
The Color Class 401
The Font and FontMetrics Classes 402
Drawing Geometric Figures 408
Drawing Lines 408
Drawing Rectangles 409
Drawing Ovals 411
Drawing Arcs 412
Drawing Polygons 414
Case Studies 417
Event-Driven Programming 421
Event and Event Source 421
Event Registration, Listening, and Handling 423
Handling Events 424
Creating Java Applications Using the Application Wizard (Optional) 433
The Application Class 436
The Frame Class 437
Modifying the Code in the Frame Class 438

**CHAPTER 11 Creating User Interfaces** 453
Introduction 454
The Component and JComponent Classes 454
Buttons 455
Labels 460
Text Fields 463
Text Areas 466
Combo Boxes 470
Lists 474
Check Boxes 477
Radio Buttons 481
CHAPTER 12 Applets and Advanced Graphics

Introduction 532

The Applet Class 532

The init method 533
The start method 533
The stop method 534
The destroy method 534

The JApplet Class 534

Creating a Java Applet Using the Applet Wizard 535

Viewing Applets 536

Viewing Applets in the Content Pane 537
Viewing Applets Using the Applet Viewer Utility 537
Viewing Applets from a Web Browser 537

The HTML File and the <applet> Tag 538

Passing Parameters to Applets 543

Enabling Applets to Run as Applications 546

Mouse Events 549

Keyboard Events 556

Modeling Dynamic Behavior Using Sequence Diagrams and Statecharts 559

Sequence Diagrams 559
Statechart Diagrams 560

Case Studies (Optional) 561

Event Adapters (Optional) 569

Standard Adapters 570
Anonymous Adapters 573

The CardLayout Manager (Optional) 576
The GridBagLayout Manager (Optional) 579
Using No Layout Manager (Optional) 583
### CHAPTER 17 Input and Output

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>730</td>
</tr>
<tr>
<td>Stream Classes</td>
<td>731</td>
</tr>
<tr>
<td>InputStream and Reader</td>
<td>732</td>
</tr>
<tr>
<td>OutputStream and Writer</td>
<td>733</td>
</tr>
<tr>
<td>The File Class and Processing External Files</td>
<td>733</td>
</tr>
<tr>
<td>Filter Streams</td>
<td>737</td>
</tr>
<tr>
<td>Data Streams</td>
<td>738</td>
</tr>
<tr>
<td>Print Streams</td>
<td>742</td>
</tr>
<tr>
<td>Buffered Streams</td>
<td>744</td>
</tr>
<tr>
<td>File Dialogs</td>
<td>748</td>
</tr>
<tr>
<td>Text Input and Output on the Console</td>
<td>753</td>
</tr>
<tr>
<td>Object Streams</td>
<td>756</td>
</tr>
<tr>
<td>The transient Keyword</td>
<td>761</td>
</tr>
<tr>
<td>Random Access Files</td>
<td>762</td>
</tr>
<tr>
<td>Parsing Text Files (Optional)</td>
<td>772</td>
</tr>
<tr>
<td>Array Streams, Piped Streams, String Streams, Pushback Streams, and Line Number Streams (Optional)</td>
<td>776</td>
</tr>
</tbody>
</table>

### CHAPTER 18 Networking

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>784</td>
</tr>
<tr>
<td>Client/Server Computing</td>
<td>784</td>
</tr>
<tr>
<td>Serving Multiple Clients and the InetAddress class</td>
<td>790</td>
</tr>
<tr>
<td>Applet Clients</td>
<td>794</td>
</tr>
<tr>
<td>Sending and Receiving Objects</td>
<td>800</td>
</tr>
<tr>
<td>The URL class and Viewing Web Pages from Applets</td>
<td>805</td>
</tr>
<tr>
<td>Retrieving Files from Web Servers</td>
<td>807</td>
</tr>
<tr>
<td>Viewing HTML Files Using JEditorPane</td>
<td>811</td>
</tr>
<tr>
<td>Cases Studies (Optional)</td>
<td>814</td>
</tr>
</tbody>
</table>

### CHAPTER 19 Java Data Structures

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>834</td>
</tr>
<tr>
<td>The Collection Interface and the AbstractCollection Class</td>
<td>835</td>
</tr>
<tr>
<td>The hashCode Method and the equals Method</td>
<td>836</td>
</tr>
<tr>
<td>The Set Interface, and the AbstractSet and HashSet Classes</td>
<td>836</td>
</tr>
<tr>
<td>The SortedSet Interface, the TreeSet Class, and the Comparator Interface</td>
<td>839</td>
</tr>
<tr>
<td>The List Interface, the AbstractList Class, and the AbstractSequentialList Class</td>
<td>842</td>
</tr>
<tr>
<td>The ArrayList and LinkedList Classes</td>
<td>843</td>
</tr>
</tbody>
</table>
The Vector and Stack Classes 845
The Map Interface, the AbstractMap Class, the SortedMap Interface, the HashMap Class, and the TreeMap Class 850
The Collections Class 853
The Arrays Class 857

APPENDIXES 863

| APPENDIX A | Java Keywords 865 |
| APPENDIX B | The ASCII Character Set 867 |
| APPENDIX C | Operator Precedence Chart 871 |
| APPENDIX D | Java Modifiers 875 |
| APPENDIX E | Number Systems and Bit Manipulations 877 |

**APPENDIX E: Number Systems and Bit Manipulations**

- Introduction 877
- Conversions between Binary Numbers and Decimal Numbers 877
- Conversions Between Hexadecimal Numbers and Decimal Numbers 878
- Conversions Between Binary Numbers and Hexadecimal Numbers 879
- Bit Manipulations 880

APPENDIX F An HTML Tutorial 883

Getting Started 883
Structure Tags 886
Text Appearance Tags 886
Content-Based Tags 886
Physical Tags 887
Paragraph-Style Tags 887
Font, Size, and Color Tags 889
List Tags 890
Ordered Lists 890
Unordered Lists 891
Definition Lists 891
Table Tags 892
Hyperlink Tags 894
Linking Documents on Different Computers 894
Linking Documents on the Same Computer 895
Jumping Within the Same Document 895
Embedding Graphics 896
Horizontal Bar Tags 896
Image Tags 897
More on HTML 898

APPENDIX G UML Graphical Notations 899
Classes and Objects 899
The Modifiers public, private, protected, and static 899
Class Relationships 900
Abstract Classes and Interfaces 901
Sequence Diagrams 901
Statechart Diagrams 901

APPENDIX H JavaBeans and Rapid Java Application Development Using JBuilder 903
Phase 1: Creating User Interface 907
Phase 2: Implementing Event Handlers 912
Phase 3: Creating Menus 915

APPENDIX I Special Floating-Point Values 921

APPENDIX J Glossary 923
Index 931