

# JAVA"

## VOLUME I-FUNDAMENTALS

EIGHTH EDITION

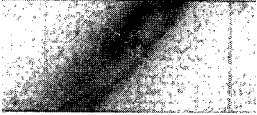


CAY S. HORSTMANN

GARY CORNELL

PREHALL<sup>CE</sup> Sun Microsystems Press

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Capetown • Sydney • Tokyo • Singapore • Mexico City



*Preface*      *xix*

*Acknowledgments*      *xxv*

## 1      AN INTRODUCTION TO JAVA

Java As a Programming Platform

The Java "White Paper" Buzzwords

Simple      3

Object Oriented      3

Network-Savvy      4

Robust      4

Secure      5

Architecture Neutral      5

Portable      6

Interpreted      6

High Performance      6

Multithreaded      7

Dynamic      7

Java Applets and the Internet	7
A Short History of Java	9
Common Misconceptions about Java	11
<b>THE JAVA PROGRAMMING ENVIRONMENT</b>	<b>15</b>
Installing the Java Development Kit	16
Downloading the JDK	16
Setting the Execution Path	18
Installing the Library Source and Documentation	20
Installing the <i>Core Java</i> Program Examples	20
Navigating the Java Directories	21
Choosing a Development Environment	21
Using the Command-Line Tools	22
Troubleshooting Hints	24
Using an Integrated Development Environment	25
Locating Compilation Errors	27
Running a Graphical Application	28
Building and Running Applets	31
<b>S</b>	<b>FUNDAMENTAL PROGRAMMING</b>
	<b>STRUCTURES IN JAVA</b>
	<b>35</b>
A Simple Java Program	36
Comments	39
Data Types	40
Integer Types	40
Floating-Point Types	41
The <code>char</code> Type	42
The <code>boolean</code> Type	44
Variables	44
Initializing Variables	45
Constants	46
Operators	46
Increment and Decrement Operators	47

Relational and boolean Operators	. 48
Bitwise Operators	49
Mathematical Functions and Constants	49
Conversions between Numeric Types	51
Casts	51
Parentheses and Operator Hierarchy	52
Enumerated Types	53
Strings	53
Substrings	54
Concatenation	54
Strings Are Immutable	54
Testing Strings for Equality	56
Code Points and Code Units	56
The String API	57
Reading the On-Line API Documentation	59
Building Strings	62
Input and Output	63
Reading Input	63
Formatting Output	65
File Input and Output	69
Control Flow	71
Block Scope	71
Conditional Statements	72
Loops	76
Determinate Loops	80
Multiple Selections—The switch Statement	84
Statements That Break Control Flow	86
Big Numbers	88
Arrays	90
The "for each" Loop	91
Array Initializers and Anonymous Arrays	92
Array Copying	93
Command-Line Parameters	95

Array Sorting	95
Multidimensional Arrays	99
Ragged Arrays	102
OBJECTS AND CLASSES	105
Introduction to Object-Oriented Programming	106
Classes	107
Objects	108
Identifying Classes	108
Relationships between Classes	109
Using Predefined Classes	111
Objects and Object Variables	111
The <code>GregorianCalendar</code> Class of the Java Library	114
Mutator and Accessor Methods	115
Defining Your Own Classes	122
An <code>Employee</code> Class	122
Use of Multiple Source Files	125
Dissecting the <code>Employee</code> Class	125
First Steps with Constructors	126
Implicit and Explicit Parameters	127
Benefits of Encapsulation	129
Class-Based Access Privileges	131
Private Methods	132
Final Instance Fields	132
Static Fields and Methods	132
Static Fields	132
Static Constants	133
Static Methods	134
Factory Methods	135
The <code>main</code> Method	135
Method Parameters	138
Object Construction	144
Overloading	144

Default Field Initialization	145	
Default Constructors	145	
Explicit Field Initialization	146	
Parameter Names	147	
Calling Another Constructor	148	
Initialization Blocks	148	
Object Destruction and the finalize Method	152	
Packages	153	
Class Importation	153	
Static Imports	155	
Addition of a Class into a Package •	155	
Package Scope	158	
The Class Path	160	
Setting the Class Path	161	
Documentation Comments	162	
Comment Insertion	162	
Class Comments	163	
Method Comments	164	
Field Comments	164	
General Comments	164	
Package and Overview Comments	165	
Comment Extraction	166	
Class Design Hints	167	
<b>5</b>	<b>INHERITANCE</b>	<b>171</b>
Classes, Superclasses, and Subclasses	172	
Inheritance Hierarchies	179	
Polymorphism	179	
Dynamic Binding	181	
Preventing Inheritance: Final Classes and Methods	183	
Casting	184	
Abstract Classes	186	
Protected Access	191	

Object: The Cosmic Superclass	192
The equals Method	193
Equality Testing and Inheritance	194
The hashCode Method	196
The toString Method	198
Generic Array Lists	204
Accessing Array List Elements	206
Compatibility between Typed and Raw Array Lists	210
Object Wrappers and Autoboxing	211
Methods with a Variable Number of Parameters	214
Enumeration Classes	215
Reflection	217
The Class Class	217
A Primer on Catching Exceptions	219
Using Reflection to Analyze the Capabilities of Classes	221
Using Reflection to Analyze Objects at Runtime	226
Using Reflection to Write Generic Array Code	231
Method Pointers!	234
Design Hints for Inheritance	238
<b>6 INTERFACES AND INNER CLASSES</b>	<b>241</b>
Interfaces	242
Properties of Interfaces	247
Interfaces and Abstract Classes	248
Object Cloning	249
Interfaces and Callbacks	255
Inner Classes	258
Use of an Inner Class to Access Object State	260
Special Syntax Rules for Inner Classes	263
Are Inner Classes Useful? Actually Necessary? Secure?	264
Local Inner Classes	266
Accessing final Variables from Outer Methods	267

	Anonymous Inner Classes	269
	Static Inner Classes	271
	Proxies	275
	Properties of Proxy Classes	279
7	<b>GRAPHICS PROGRAMMING</b>	<b>281</b>
	Introducing Swing	282
	Creating a Frame	285
	Positioning a Frame	288
	Frame Properties	290
	Determining a Good Frame Size	290
	Displaying Information in a Component	294
	Working with 2D Shapes	299
	Using Color	307
	Using Special Fonts for Text	310
	Displaying Images	318
8	<b>EVENT HANDLING</b>	<b>323</b>
	Basics of Event Handling	324
	Example: Handling a Button Click	326
	Becoming Comfortable with Inner Classes	331
	Creating Listeners Containing a Single Method Call	334
	Example: Changing the Look and Feel	334
	Adapter Classes	338
	Actions	342
	Mouse Events	349
	The AWT Event Hierarchy	357
	Semantic and Low-Level Events	358
9	<b>USER INTERFACE COMPONENTS WITH SWING</b>	<b>361</b>
	Swing and the Model-View-Controller Design Pattern	362
	Design Patterns	362



The Model-View-Controller Pattern	363
A Model-View-Controller Analysis of Swing Buttons	367
Introduction to Layout Management	368
Border Layout	371
Grid Layout	373
Text Input	377
Text Fields	378
Labels and Labeling Components	379
Password Fields	381
Text Areas	381
Scroll Panes	382
Choice Components	385
Checkboxes	385
Radio Buttons	388
Borders	392
Combo Boxes	396
Sliders	400
Menus	406
Menu Building	406
Icons in Menu Items	408
Checkbox and Radio Button Menu Items	409
Pop-Up Menus	410
Keyboard Mnemonics and Accelerators	412
Enabling and Disabling Menu Items	414
Toolbars	419
Tooltips	421
Sophisticated Layout Management	424
The Grid Bag Layout	426
Group Layout	436
Using No Layout Manager	445
Custom Layout Managers	446
Traversal Order	450

Dialog Boxes	452	
Option Dialogs	453	
Creating Dialogs	462	
Data Exchange	467	
File Dialogs	474	
Color Choosers	485	
<b>10</b>	<b>DEPLOYING APPLICATIONS AND APPLETS</b>	<b>493</b>
JAR Files	494	
The Manifest	495	
Executable JAR Files	496	
Resources	497	
Sealing	500	
Java Web Start	501	
The Sandbox	504	
Signed Code	505	
TheJNLPAPI	507	
Applets	516	
A Simple Applet	516	
The Applet HTML Tag and Its Attributes	521	
The object Tag	524	
Use of Parameters to Pass Information to Applets	524	
Accessing Image and Audio Files	530	
The Applet Context	531	
Storage of Application Preferences	539	
Property Maps	539	
The Preferences API	544	
<b>11</b>	<b>EXCEPTIONS, LOGGING, ASSERTIONS, AND DEBUGGING</b>	<b>551</b>
Dealing with Errors	552	
The Classification of Exceptions	553	
Declaring Checked Exceptions	555	

	How to Throw an Exception	557
	Creating Exception Classes	558
	Catching Exceptions	559
	Catching Multiple Exceptions	561
	Rethrowing and Chaining Exceptions	562
	The finally Clause	563
	Analyzing Stack Trace Elements	566
	Tips for Using Exceptions	568
	Using Assertions	571
	Assertion Enabling and Disabling	572
	Using Assertions for Parameter Checking	573
	Using Assertions for Documenting Assumptions	574
	Logging	575
	Basic Logging	575
	Advanced Logging	576
	Changing the Log Manager Configuration	578
	Localization	579
	Handlers	579
	Filters	583
	Formatters	583
	A Logging Recipe	583
	Debugging Tips	591
	Using a Console Window	597
	Tracing AWT Events	598
	Letting the AWT Robot Do the Work	603
	Using a Debugger	607
12	GENERIC PROGRAMMING	613
	Why Generic Programming?	614
	Who Wants to Be a Generic Programmer?	615
	Definition of a Simple Generic Class	616
	Generic Methods	618
	Bounds for Type Variables	619

Generic Code and the Virtual Machine	621
Translating Generic Expressions	622
Translating Generic Methods	623
Calling Legacy Code	625
Restrictions and Limitations	626
Type Parameters Cannot Be Instantiated with Primitive Types	626
Runtime Type Inquiry Only Works with Raw Types	626
You Cannot Throw or Catch Instances of a Generic Class	626
Arrays of Parameterized Types Are Not Legal	627
You Cannot Instantiate Type Variables	628
Type Variables Are Not Valid in Static Contexts of Generic Classes	629
Beware of Clashes After Erasure	629
Inheritance Rules for Generic Types	630
Wildcard Types	632
Supertype Bounds for Wildcards	634
Unbounded Wildcards	636
Wildcard Capture	636
Reflection and Generics	640
Using <code>Class&lt;T&gt;</code> Parameters for Type Matching	641
Generic Type Information in the Virtual Machine	641
<b>13 COLLECTIONS</b>	<b>649</b>
Collection Interfaces	650
Separating Collection Interfaces and Implementation	650
Collection and Iterator Interfaces in the Java Library	653
Concrete Collections	658
Linked Lists	659
Array Lists	668
Hash Sets	668
Tree Sets	672

	Object Comparison	673
	Queues and Deques	678
	Priority Queues	679
	Maps	680
	Specialized Set and Map Classes	685
	The Collections Framework	689
	Views and Wrappers	693
	Bulk Operations	699
	Converting between Collections and Arrays	700
	Algorithms	700
	Sorting and Shuffling	701
	Binary Search	704
	Simple Algorithms	705
	Writing Your Own Algorithms	706
	Legacy Collections	707
	The <code>Hashtable</code> Class	708
	Enumerations	708
	Property Maps	709
	Stacks	709
	Bit Sets	710
<b>14</b>	<b>MULTITHREADING</b>	<b>715</b>
	What Are Threads?	716
	Using Threads to Give Other Tasks a Chance	722
	Interrupting Threads	728
	Thread States	730
	New Threads	731
	Runnable Threads	731
	Blocked and Waiting Threads	731
	Terminated Threads	733
	Thread Properties	733
	Thread Priorities	733
	Daemon Threads	734

Handlers for Uncaught Exceptions	735
Synchronization	736
An Example of a Race Condition	736
The Race Condition Explained	740
Lock Objects	742
Condition Objects	745
The <code>synchronized</code> Keyword	750
Synchronized Blocks	754
The Monitor Concept	755
Volatile Fields	756
Deadlocks	757
Lock Testing and Timeouts	760
ReadAA/rite Locks	761
Why the <code>stop</code> and <code>suspend</code> Methods Are Deprecated	762
Blocking Queues	764
Thread-Safe Collections	771
Efficient Maps, Sets, and Queues	771
Copy on Write Arrays	773
Older Thread-Safe Collections	773
Callables and Futures	774
Executors	778
Thread Pools	779
Scheduled Execution	783
Controlling Groups of Tasks	784
Synchronizers	785
Semaphores	786
Countdown Latches	786
Barriers	787
Exchangers	787
Synchronous Queues	787
Example: Pausing and Resuming an Animation	• 788
Threads and Swing	794
Running Time-Consuming Tasks	795

Using the Swing Worker	799
The Single-Thread Rule	806

*Appendix* 809

*Index* 813