

# Java™ for Programmers

Second Edition

Contains 200+  
Examples

Covers  
Java SE 7

Java™ SE 7 • Java APIs • Object-Oriented Programming  
Database • SQL • JDBC™ • JavaDB/ Apache Derby/MySQL  
Networking • JavaServer™ Faces 2.0 • AJAX-Enabled Web Applications  
Web Services • Generics • Collections • Files • Exception Handling  
Multithreading • Swing Graphical User Interfaces • Graphics/Java 2D™  
Multimedia • OOD/UML® ATM Case Study • Debugger  
Online Introduction to Android™ App Development

PAUL DEITEL • HARVEY DEITEL

---

**JAVA™ FOR PROGRAMMERS**  
**SECOND EDITION**  
**DEITEL® DEVELOPER SERIES**

---

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales  
(800) 382-3419  
corpsales@pearsontechgroup.com

For sales outside the U. S., please contact:

International Sales  
international@pearsoned.com

Visit us on the Web: [informit.com/ph](http://informit.com/ph)

*Library of Congress Cataloging-in-Publication Data*

On file

© 2012 Pearson Education, Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax (617) 671-3447

ISBN-13: 978-0-13282154-4

ISBN-10: 0-13-282154-0

Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.

First printing, March 2012

---

# JAVA™ FOR PROGRAMMERS

## SECOND EDITION

### DEITEL® DEVELOPER SERIES

---

Paul Deitel  
*Deitel & Associates, Inc.*

Harvey Deitel  
*Deitel & Associates, Inc.*



PRENTICE  
HALL

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

---

## Trademarks

DEITEL, the double-thumbs-up bug and DIVE INTO are registered trademarks of Deitel and Associates, Inc.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Microsoft, Internet Explorer and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group.

Apache is a trademark of The Apache Software Foundation.

CSS, XHTML and XML are registered trademarks of the World Wide Web Consortium.

Firefox is a registered trademark of the Mozilla Foundation.

Google is a trademark of Google, Inc.

Web 2.0 is a service mark of CMP Media.

Throughout this book, trademarks are used. Rather than put a trademark symbol in every occurrence of a trademarked name, we state that we are using the names in an editorial fashion only and to the benefit of the trademark owner, with no intention of infringement of the trademark.

---

*In memory of Clifford "Spike" Stephens,  
A dear friend who will be greatly missed.*

*Paul and Harvey Deitel*

---

*This page intentionally left blank*

# Contents

<b>Preface</b>	<b>xxi</b>
<b>Before You Begin</b>	<b>xxix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction	2
1.2 Introduction to Object Technology	2
1.3 Open Source Software	5
1.4 Java and a Typical Java Development Environment	7
1.5 Test-Driving a Java Application	11
1.6 Web 2.0: Going Social	15
1.7 Software Technologies	18
1.8 Keeping Up to Date with Information Technologies	20
1.9 Wrap-Up	21
<b>2 Introduction to Java Applications</b>	<b>22</b>
2.1 Introduction	23
2.2 Your First Program in Java: Printing a Line of Text	23
2.3 Modifying Your First Java Program	27
2.4 Displaying Text with <code>printf</code>	29
2.5 Another Application: Adding Integers	30
2.6 Arithmetic	34
2.7 Decision Making: Equality and Relational Operators	35
2.8 Wrap-Up	38
<b>3 Introduction to Classes, Objects, Methods and Strings</b>	<b>39</b>
3.1 Introduction	40
3.2 Declaring a Class with a Method and Instantiating an Object of a Class	40
3.3 Declaring a Method with a Parameter	44
3.4 Instance Variables, <i>set</i> Methods and <i>get</i> Methods	47
3.5 Primitive Types vs. Reference Types	52
3.6 Initializing Objects with Constructors	53



3.7	Floating-Point Numbers and Type <code>double</code>	56
3.8	Wrap-Up	60
<b>4</b>	<b>Control Statements: Part I</b>	<b>61</b>
4.1	Introduction	62
4.2	Control Structures	62
4.3	<code>if</code> Single-Selection Statement	64
4.4	<code>if...else</code> Double-Selection Statement	65
4.5	<code>while</code> Repetition Statement	68
4.6	Counter-Controlled Repetition	70
4.7	Sentinel-Controlled Repetition	73
4.8	Nested Control Statements	78
4.9	Compound Assignment Operators	81
4.10	Increment and Decrement Operators	82
4.11	Primitive Types	85
4.12	Wrap-Up	85
<b>5</b>	<b>Control Statements: Part 2</b>	<b>86</b>
5.1	Introduction	87
5.2	Essentials of Counter-Controlled Repetition	87
5.3	<code>for</code> Repetition Statement	89
5.4	Examples Using the <code>for</code> Statement	92
5.5	<code>do...while</code> Repetition Statement	96
5.6	<code>switch</code> Multiple-Selection Statement	98
5.7	<code>break</code> and <code>continue</code> Statements	105
5.8	Logical Operators	107
5.9	Wrap-Up	113
<b>6</b>	<b>Methods: A Deeper Look</b>	<b>114</b>
6.1	Introduction	115
6.2	Program Modules in Java	115
6.3	<code>static</code> Methods, <code>static</code> Fields and Class <code>Math</code>	115
6.4	Declaring Methods with Multiple Parameters	118
6.5	Notes on Declaring and Using Methods	121
6.6	Argument Promotion and Casting	122
6.7	Java API Packages	123
6.8	Case Study: Random-Number Generation	125
	6.8.1 Generalized Scaling and Shifting of Random Numbers	129
	6.8.2 Random-Number Repeatability for Testing and Debugging	129
6.9	Case Study: A Game of Chance; Introducing Enumerations	130
6.10	Scope of Declarations	134
6.11	Method Overloading	137
6.12	Wrap-Up	139

<b>7</b>	<b>Arrays and ArrayLists</b>	<b>140</b>
7.1	Introduction	141
7.2	Arrays	141
7.3	Declaring and Creating Arrays	143
7.4	Examples Using Arrays	144
7.5	Case Study: Card Shuffling and Dealing Simulation	153
7.6	Enhanced <code>for</code> Statement	157
7.7	Passing Arrays to Methods	159
7.8	Case Study: Class <code>GradeBook</code> Using an Array to Store Grades	162
7.9	Multidimensional Arrays	167
7.10	Case Study: Class <code>GradeBook</code> Using a Two-Dimensional Array	171
7.11	Variable-Length Argument Lists	177
7.12	Using Command-Line Arguments	178
7.13	Class Arrays	180
7.14	Introduction to Collections and Class <code>ArrayList</code>	183
7.15	Wrap-Up	186
<b>8</b>	<b>Classes and Objects: A Deeper Look</b>	<b>187</b>
8.1	Introduction	188
8.2	Time Class Case Study	188
8.3	Controlling Access to Members	192
8.4	Referring to the Current Object's Members with the <code>this</code> Reference	193
8.5	Time Class Case Study: Overloaded Constructors	195
8.6	Default and No-Argument Constructors	201
8.7	Notes on <i>Set</i> and <i>Get</i> Methods	202
8.8	Composition	203
8.9	Enumerations	206
8.10	Garbage Collection and Method <code>finalize</code>	209
8.11	<code>static</code> Class Members	210
8.12	<code>static</code> Import	213
8.13	<code>final</code> Instance Variables	214
8.14	Time Class Case Study: Creating Packages	215
8.15	Package Access	221
8.16	Wrap-Up	222
<b>9</b>	<b>Object-Oriented Programming: Inheritance</b>	<b>224</b>
9.1	Introduction	225
9.2	Superclasses and Subclasses	226
9.3	<code>protected</code> Members	228
9.4	Relationship between Superclasses and Subclasses	228
9.4.1	Creating and Using a <code>CommissionEmployee</code> Class	229
9.4.2	Creating and Using a <code>BasePlusCommissionEmployee</code> Class	235
9.4.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	240

9.4.4	CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables	242
9.4.5	CommissionEmployee–BasePlusCommissionEmployee Inheritance Hierarchy Using private Instance Variables	245
9.5	Constructors in Subclasses	250
9.6	Software Engineering with Inheritance	251
9.7	Class Object	252
9.8	Wrap-Up	253

## **10 Object-Oriented Programming: Polymorphism 254**

10.1	Introduction	255
10.2	Polymorphism Examples	257
10.3	Demonstrating Polymorphic Behavior	258
10.4	Abstract Classes and Methods	260
10.5	Case Study: Payroll System Using Polymorphism	262
10.5.1	Abstract Superclass Employee	263
10.5.2	Concrete Subclass SalariedEmployee	266
10.5.3	Concrete Subclass HourlyEmployee	268
10.5.4	Concrete Subclass CommissionEmployee	270
10.5.5	Indirect Concrete Subclass BasePlusCommissionEmployee	271
10.5.6	Polymorphic Processing, Operator instanceof and Downcasting	273
10.5.7	Summary of the Allowed Assignments Between Superclass and Subclass Variables	277
10.6	final Methods and Classes	278
10.7	Case Study: Creating and Using Interfaces	279
10.7.1	Developing a Payable Hierarchy	280
10.7.2	Interface Payable	281
10.7.3	Class Invoice	282
10.7.4	Modifying Class Employee to Implement Interface Payable	284
10.7.5	Modifying Class SalariedEmployee for Use in the Payable Hierarchy	286
10.7.6	Using Interface Payable to Process Invoices and Employees Polymorphically	288
10.7.7	Common Interfaces of the Java API	289
10.8	Wrap-Up	290

## **11 Exception Handling: A Deeper Look 292**

11.1	Introduction	293
11.2	Example: Divide by Zero without Exception Handling	293
11.3	Example: Handling ArithmeticExceptions and InputMismatchExceptions	296
11.4	When to Use Exception Handling	301
11.5	Java Exception Hierarchy	301
11.6	finally Block	304
11.7	Stack Unwinding and Obtaining Information from an Exception Object	308

11.8	Chained Exceptions	311
11.9	Declaring New Exception Types	313
11.10	Preconditions and Postconditions	314
11.11	Assertions	315
11.12	(New in Java SE 7) Multi-catch: Handling Multiple Exceptions in One catch	316
11.13	(New in Java SE 7) try-with-Resources: Automatic Resource Deallocation	316
11.14	Wrap-Up	317

## **12 ATM Case Study, Part 1: Object-Oriented Design with the UML** **318**

12.1	Case Study Introduction	319
12.2	Examining the Requirements Document	319
12.3	Identifying the Classes in a Requirements Document	327
12.4	Identifying Class Attributes	333
12.5	Identifying Objects' States and Activities	338
12.6	Identifying Class Operations	342
12.7	Indicating Collaboration Among Objects	348
12.8	Wrap-Up	355

## **13 ATM Case Study Part 2: Implementing an Object-Oriented Design** **359**

13.1	Introduction	360
13.2	Starting to Program the Classes of the ATM System	360
13.3	Incorporating Inheritance and Polymorphism into the ATM System	365
13.4	ATM Case Study Implementation	371
13.4.1	Class ATM	372
13.4.2	Class Screen	377
13.4.3	Class Keypad	378
13.4.4	Class CashDispenser	379
13.4.5	Class DepositSlot	380
13.4.6	Class Account	381
13.4.7	Class BankDatabase	383
13.4.8	Class Transaction	386
13.4.9	Class BalanceInquiry	387
13.4.10	Class Withdrawal	388
13.4.11	Class Deposit	392
13.4.12	Class ATMCaseStudy	395
13.5	Wrap-Up	395

## **14 GUI Components: Part I** **398**

14.1	Introduction	399
14.2	Java's New Nimbus Look-and-Feel	400

14.3	Simple GUI-Based Input/Output with <code>JOptionPane</code>	401
14.4	Overview of Swing Components	404
14.5	Displaying Text and Images in a Window	406
14.6	Text Fields and an Introduction to Event Handling with Nested Classes	410
14.7	Common GUI Event Types and Listener Interfaces	416
14.8	How Event Handling Works	418
14.9	<code>JButton</code>	420
14.10	Buttons That Maintain State	423
	14.10.1 <code>JCheckBox</code>	423
	14.10.2 <code>JRadioButton</code>	426
14.11	<code>JComboBox</code> ; Using an Anonymous Inner Class for Event Handling	429
14.12	<code>JList</code>	433
14.13	Multiple-Selection Lists	435
14.14	Mouse Event Handling	438
14.15	Adapter Classes	443
14.16	<code>JPanel</code> Subclass for Drawing with the Mouse	446
14.17	Key Event Handling	450
14.18	Introduction to Layout Managers	453
	14.18.1 <code>FlowLayout</code>	454
	14.18.2 <code>BorderLayout</code>	457
	14.18.3 <code>GridLayout</code>	460
14.19	Using Panels to Manage More Complex Layouts	462
14.20	<code>JTextArea</code>	464
14.21	Wrap-Up	467

## **15 Graphics and Java 2D** **468**

15.1	Introduction	469
15.2	Graphics Contexts and Graphics Objects	471
15.3	Color Control	472
15.4	Manipulating Fonts	479
15.5	Drawing Lines, Rectangles and Ovals	484
15.6	Drawing Arcs	488
15.7	Drawing Polygons and Polylines	491
15.8	Java 2D API	494
15.9	Wrap-Up	501

## **16 Strings, Characters and Regular Expressions** **502**

16.1	Introduction	503
16.2	Fundamentals of Characters and Strings	503
16.3	Class <code>String</code>	504
	16.3.1 <code>String</code> Constructors	504
	16.3.2 <code>String</code> Methods <code>length</code> , <code>charAt</code> and <code>getChars</code>	505
	16.3.3 Comparing Strings	506
	16.3.4 Locating Characters and Substrings in Strings	511

16.3.5	Extracting Substrings from Strings	513
16.3.6	Concatenating Strings	514
16.3.7	Miscellaneous String Methods	514
16.3.8	String Method <code>valueOf</code>	516
16.4	Class <code>StringBuilder</code>	517
16.4.1	<code>StringBuilder</code> Constructors	518
16.4.2	<code>StringBuilder</code> Methods <code>length</code> , <code>capacity</code> , <code>setLength</code> and <code>ensureCapacity</code>	518
16.4.3	<code>StringBuilder</code> Methods <code>charAt</code> , <code>setCharAt</code> , <code>getChars</code> and <code>reverse</code>	520
16.4.4	<code>StringBuilder</code> append Methods	521
16.4.5	<code>StringBuilder</code> Insertion and Deletion Methods	523
16.5	Class <code>Character</code>	524
16.6	Tokenizing Strings	529
16.7	Regular Expressions, Class <code>Pattern</code> and Class <code>Matcher</code>	530
16.8	Wrap-Up	538

## **17 Files, Streams and Object Serialization** **539**

17.1	Introduction	540
17.2	Files and Streams	540
17.3	Class <code>File</code>	542
17.4	Sequential-Access Text Files	546
17.4.1	Creating a Sequential-Access Text File	546
17.4.2	Reading Data from a Sequential-Access Text File	553
17.4.3	Case Study: A Credit-Inquiry Program	556
17.4.4	Updating Sequential-Access Files	561
17.5	Object Serialization	562
17.5.1	Creating a Sequential-Access File Using Object Serialization	563
17.5.2	Reading and Deserializing Data from a Sequential-Access File	569
17.6	Additional <code>java.io</code> Classes	571
17.6.1	Interfaces and Classes for Byte-Based Input and Output	571
17.6.2	Interfaces and Classes for Character-Based Input and Output	573
17.7	Opening Files with <code>JFileChooser</code>	574
17.8	Wrap-Up	577

## **18 Generic Collections** **578**

18.1	Introduction	579
18.2	Collections Overview	579
18.3	Type-Wrapper Classes for Primitive Types	580
18.4	Autoboxing and Auto-Unboxing	581
18.5	Interface <code>Collection</code> and Class <code>Collections</code>	581
18.6	Lists	582
18.6.1	<code>ArrayList</code> and <code>Iterator</code>	583
18.6.2	<code>LinkedList</code>	585

18.7	Collections Methods	590
18.7.1	Method <code>sort</code>	591
18.7.2	Method <code>shuffle</code>	594
18.7.3	Methods <code>reverse</code> , <code>fill</code> , <code>copy</code> , <code>max</code> and <code>min</code>	596
18.7.4	Method <code>binarySearch</code>	598
18.7.5	Methods <code>addAll</code> , <code>frequency</code> and <code>disjoint</code>	600
18.8	Stack Class of Package <code>java.util</code>	602
18.9	Class <code>PriorityQueue</code> and Interface <code>Queue</code>	604
18.10	Sets	605
18.11	Maps	608
18.12	<code>Properties</code> Class	612
18.13	Synchronized Collections	615
18.14	Unmodifiable Collections	615
18.15	Abstract Implementations	616
18.16	Wrap-Up	616

## **19 Generic Classes and Methods** **618**

19.1	Introduction	619
19.2	Motivation for Generic Methods	619
19.3	Generic Methods: Implementation and Compile-Time Translation	622
19.4	Additional Compile-Time Translation Issues: Methods That Use a Type Parameter as the Return Type	625
19.5	Overloading Generic Methods	628
19.6	Generic Classes	628
19.7	Raw Types	636
19.8	Wildcards in Methods That Accept Type Parameters	640
19.9	Generics and Inheritance: Notes	644
19.10	Wrap-Up	645

## **20 Applets and Java Web Start** **646**

20.1	Introduction	647
20.2	Sample Applets Provided with the JDK	648
20.3	Simple Java Applet: Drawing a String	652
20.3.1	Executing <code>WelcomeApplet</code> in the <code>appletviewer</code>	654
20.3.2	Executing an Applet in a Web Browser	656
20.4	Applet Life-Cycle Methods	656
20.5	Initialization with Method <code>init</code>	657
20.6	Sandbox Security Model	659
20.7	Java Web Start and the Java Network Launch Protocol (JNLP)	661
20.7.1	Packaging the <code>DrawTest</code> Applet for Use with Java Web Start	661
20.7.2	JNLP Document for the <code>DrawTest</code> Applet	662
20.8	Wrap-Up	666

<b>21</b>	<b>Multimedia: Applets and Applications</b>	<b>667</b>
21.1	Introduction	668
21.2	Loading, Displaying and Scaling Images	669
21.3	Animating a Series of Images	675
21.4	Image Maps	682
21.5	Loading and Playing Audio Clips	685
21.6	Playing Video and Other Media with Java Media Framework	688
21.7	Wrap-Up	692
21.8	Web Resources	692
<b>22</b>	<b>GUI Components: Part 2</b>	<b>694</b>
22.1	Introduction	695
22.2	JSlider	695
22.3	Windows: Additional Notes	699
22.4	Using Menus with Frames	700
22.5	JPopupMenu	708
22.6	Pluggable Look-and-Feel	711
22.7	JDesktopPane and JInternalFrame	716
22.8	JTabbedPane	720
22.9	Layout Managers: BorderLayout and GridBagLayout	722
22.10	Wrap-Up	734
<b>23</b>	<b>Multithreading</b>	<b>735</b>
23.1	Introduction	736
23.2	Thread States: Life Cycle of a Thread	738
23.3	Creating and Executing Threads with Executor Framework	741
23.4	Thread Synchronization	744
	23.4.1 Unsynchronized Data Sharing	745
	23.4.2 Synchronized Data Sharing—Making Operations Atomic	749
23.5	Producer/Consumer Relationship without Synchronization	752
23.6	Producer/Consumer Relationship: ArrayBlockingQueue	760
23.7	Producer/Consumer Relationship with Synchronization	763
23.8	Producer/Consumer Relationship: Bounded Buffers	769
23.9	Producer/Consumer Relationship: The Lock and Condition Interfaces	776
23.10	Concurrent Collections Overview	783
23.11	Multithreading with GUI	785
	23.11.1 Performing Computations in a Worker Thread	786
	23.11.2 Processing Intermediate Results with SwingWorker	792
23.12	Interfaces Callable and Future	799
23.13	Java SE 7: Fork/Join Framework	799
23.14	Wrap-Up	800



<b>24</b>	<b>Networking</b>	<b>801</b>
24.1	Introduction	802
24.2	Manipulating URLs	803
24.3	Reading a File on a Web Server	808
24.4	Establishing a Simple Server Using Stream Sockets	811
24.5	Establishing a Simple Client Using Stream Sockets	813
24.6	Client/Server Interaction with Stream Socket Connections	813
24.7	Datagrams: Connectionless Client/Server Interaction	825
24.8	Client/Server Tic-Tac-Toe Using a Multithreaded Server	833
24.9	[Web Bonus] Case Study: DeitelMessenger	848
24.10	Wrap-Up	848
<b>25</b>	<b>Accessing Databases with JDBC</b>	<b>849</b>
25.1	Introduction	850
25.2	Relational Databases	851
25.3	Relational Database Overview: The books Database	852
25.4	SQL	855
25.4.1	Basic SELECT Query	856
25.4.2	WHERE Clause	857
25.4.3	ORDER BY Clause	859
25.4.4	Merging Data from Multiple Tables: INNER JOIN	860
25.4.5	INSERT Statement	862
25.4.6	UPDATE Statement	863
25.4.7	DELETE Statement	864
25.5	Instructions for Installing MySQL and MySQL Connector/J	864
25.6	Instructions for Setting Up a MySQL User Account	865
25.7	Creating Database books in MySQL	866
25.8	Manipulating Databases with JDBC	867
25.8.1	Connecting to and Querying a Database	867
25.8.2	Querying the books Database	872
25.9	RowSet Interface	885
25.10	Java DB/Apache Derby	887
25.11	PreparedStatement	889
25.12	Stored Procedures	904
25.13	Transaction Processing	905
25.14	Wrap-Up	905
25.15	Web Resources	906
<b>26</b>	<b>JavaServer™ Faces Web Apps: Part I</b>	<b>907</b>
26.1	Introduction	908
26.2	HyperText Transfer Protocol (HTTP) Transactions	909
26.3	Multitier Application Architecture	912
26.4	Your First JSF Web App	913
26.4.1	The Default index.xhtml Document: Introducing Facelets	914

26.4.2	Examining the <code>WebTimeBean</code> Class	916
26.4.3	Building the <code>WebTime</code> JSF Web App in NetBeans	918
26.5	Model-View-Controller Architecture of JSF Apps	922
26.6	Common JSF Components	922
26.7	Validation Using JSF Standard Validators	926
26.8	Session Tracking	933
26.8.1	Cookies	934
26.8.2	Session Tracking with <code>@SessionScoped</code> Beans	935
26.9	Wrap-Up	941
<b>27</b>	<b>JavaServer™ Faces Web Apps: Part 2</b>	<b>942</b>
27.1	Introduction	943
27.2	Accessing Databases in Web Apps	943
27.2.1	Setting Up the Database	945
27.2.2	<code>@ManagedBean</code> Class <code>AddressBean</code>	948
27.2.3	<code>index.xhtml</code> Facelets Page	952
27.2.4	<code>addentry.xhtml</code> Facelets Page	954
27.3	Ajax	956
27.4	Adding Ajax Functionality to the Validation App	958
27.5	Wrap-Up	961
<b>28</b>	<b>Web Services</b>	<b>962</b>
28.1	Introduction	963
28.2	Web Service Basics	965
28.3	Simple Object Access Protocol (SOAP)	965
28.4	Representational State Transfer (REST)	965
28.5	JavaScript Object Notation (JSON)	966
28.6	Publishing and Consuming SOAP-Based Web Services	966
28.6.1	Creating a Web Application Project and Adding a Web Service Class in NetBeans	966
28.6.2	Defining the <code>WelcomeSOAP</code> Web Service in NetBeans	967
28.6.3	Publishing the <code>WelcomeSOAP</code> Web Service from NetBeans	970
28.6.4	Testing the <code>WelcomeSOAP</code> Web Service with GlassFish Application Server's Tester Web Page	971
28.6.5	Describing a Web Service with the Web Service Description Language (WSDL)	972
28.6.6	Creating a Client to Consume the <code>WelcomeSOAP</code> Web Service	973
28.6.7	Consuming the <code>WelcomeSOAP</code> Web Service	975
28.7	Publishing and Consuming REST-Based XML Web Services	978
28.7.1	Creating a REST-Based XML Web Service	978
28.7.2	Consuming a REST-Based XML Web Service	981
28.8	Publishing and Consuming REST-Based JSON Web Services	983
28.8.1	Creating a REST-Based JSON Web Service	983
28.8.2	Consuming a REST-Based JSON Web Service	985

28.9	Session Tracking in a SOAP Web Service	987
28.9.1	Creating a Blackjack Web Service	988
28.9.2	Consuming the Blackjack Web Service	991
28.10	Consuming a Database-Driven SOAP Web Service	1002
28.10.1	Creating the Reservation Database	1003
28.10.2	Creating a Web Application to Interact with the Reservation Service	1006
28.11	Equation Generator: Returning User-Defined Types	1009
28.11.1	Creating the EquationGeneratorXML Web Service	1012
28.11.2	Consuming the EquationGeneratorXML Web Service	1013
28.11.3	Creating the EquationGeneratorJSON Web Service	1017
28.11.4	Consuming the EquationGeneratorJSON Web Service	1017
28.12	Wrap-Up	1020
<b>A</b>	<b>Operator Precedence Chart</b>	<b>1022</b>
<b>B</b>	<b>ASCII Character Set</b>	<b>1024</b>
<b>C</b>	<b>Keywords and Reserved Words</b>	<b>1025</b>
<b>D</b>	<b>Primitive Types</b>	<b>1026</b>
<b>E</b>	<b>Using the Java API Documentation</b>	<b>1027</b>
E.1	Introduction	1027
E.2	Navigating the Java API	1028
<b>F</b>	<b>Using the Debugger</b>	<b>1036</b>
F.1	Introduction	1037
F.2	Breakpoints and the run, stop, cont and print Commands	1037
F.3	The print and set Commands	1041
F.4	Controlling Execution Using the step, step up and next Commands	1043
F.5	The watch Command	1046
F.6	The clear Command	1049
F.7	Wrap-Up	1051
<b>G</b>	<b>Formatted Output</b>	<b>1052</b>
G.1	Introduction	1053
G.2	Streams	1053
G.3	Formatting Output with printf	1053

G.4	Printing Integers	1054
G.5	Printing Floating-Point Numbers	1055
G.6	Printing Strings and Characters	1057
G.7	Printing Dates and Times	1058
G.8	Other Conversion Characters	1060
G.9	Printing with Field Widths and Precisions	1062
G.10	Using Flags in the printf Format String	1064
G.11	Printing with Argument Indices	1068
G.12	Printing Literals and Escape Sequences	1068
G.13	Formatting Output with Class Formatter	1069
G.14	Wrap-Up	1070
<b>H</b>	<b>GroupLayout</b>	<b>1071</b>
H.1	Introduction	1071
H.2	GroupLayout Basics	1071
H.3	Building a ColorChooser	1072
H.4	GroupLayout Web Resources	1082
<b>I</b>	<b>Java Desktop Integration Components</b>	<b>1083</b>
I.1	Introduction	1083
I.2	Splash Screens	1083
I.3	Desktop Class	1085
I.4	Tray Icons	1087
<b>J</b>	<b>UML 2: Additional Diagram Types</b>	<b>1089</b>
J.1	Introduction	1089
J.2	Additional Diagram Types	1089
	<b>Index</b>	<b>1091</b>

---

*This page intentionally left blank*

# Preface

*Live in fragments no longer, only connect.*

—Edgar Morgan Foster

Welcome to Java and *Java for Programmers, Second Edition*! This book presents leading-edge computing technologies for software developers.

We focus on software engineering best practices. At the heart of the book is the Deitel signature “live-code approach”—concepts are presented in the context of complete working programs, rather than in code snippets. Each complete code example is accompanied by live sample executions. All the source code is available at

[www.deitel.com/books/javafp2/](http://www.deitel.com/books/javafp2/)

As you read the book, if you have questions, send an e-mail to [deitel@deitel.com](mailto:deitel@deitel.com); we'll respond promptly. For updates on this book, visit the website shown above, follow us on Facebook ([www.facebook.com/DeitelFan](http://www.facebook.com/DeitelFan)) and Twitter (@deitel), and subscribe to the *Deitel® Buzz Online* newsletter ([www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)).

## Features

Here are the key features of *Java for Programmers, 2/e*:

### *Java Standard Edition (SE) 7*

- ***Easy to use as a Java SE 6 or Java SE 7 book.*** We cover the new Java SE 7 features in modular sections. Here's some of the new functionality: Strings in switch statements, the try-with-resources statement for managing AutoClosable objects, multi-catch for defining a single exception handler to replace multiple exception handlers that perform the same task and inferring the types of generic objects from the variable they're assigned to by using the <> notation. We also overview the new concurrency API features.
- ***Java SE 7's AutoClosable versions of Connection, Statement and ResultSet.*** With the source code for Chapter 25, Accessing Databases with JDBC, we provide a version of the chapter's first example that's implemented using Java SE 7's AutoClosable versions of Connection, Statement and ResultSet. AutoClosable objects reduce the likelihood of resource leaks when you use them with Java SE 7's try-with-resources statement, which automatically closes the AutoClosable objects allocated in the parentheses following the try keyword.

### *Object Technology*

- ***Object-oriented programming and design.*** We review the basic concepts and terminology of object technology in Chapter 1. Readers develop their first customized classes and objects in Chapter 3.

- *Exception handling.* We integrate basic exception handling early in the book and cover it in detail in Chapter 11, Exception Handling: A Deeper Look.
- *Class Arrays and ArrayList.* Chapter 7 covers class Arrays—which contains methods for performing common array manipulations—and class ArrayList—which implements a dynamically resizable array-like data structure.
- *OO case studies.* The early classes and objects presentation features Time, Employee and GradeBook class case studies that weave their way through multiple sections and chapters, gradually introducing deeper OO concepts.
- *Case Study: Using the UML to Develop an Object-Oriented Design and Java Implementation of an ATM.* The UML™ (Unified Modeling Language™) is the industry-standard graphical language for modeling object-oriented systems. Chapters 12–13 include a case study on object-oriented design using the UML. We design and implement the software for a simple automated teller machine (ATM). We analyze a typical requirements document that specifies the system to be built. We determine the classes needed to implement that system, the attributes the classes need to have, the behaviors the classes need to exhibit and specify how the classes must interact with one another to meet the system requirements. From the design we produce a *complete* Java implementation. Readers often report having a “light-bulb moment”—the case study helps them “tie it all together” and really understand object orientation in Java.
- *Reordered generics presentation.* We begin with generic class ArrayList in Chapter 7. Because *you’ll understand basic generics concepts early in the book*, our later data structures discussions provide a deeper treatment of generic collections—showing how to use the built-in collections of the Java API. We then show how to implement generic methods and classes.

#### *Database and Web Development*

- *JDBC 4.* Chapter 25, Accessing Databases with JDBC, covers JDBC 4 and uses the Java DB/Apache Derby and MySQL database management systems. The chapter features an OO case study on developing a database-driven address book that demonstrates prepared statements and JDBC 4’s automatic driver discovery.
- *Java Server Faces (JSF) 2.0.* Chapters 26–27 have been updated with JavaServer Faces (JSF) 2.0 technology, which greatly simplifies building JSF web applications. Chapter 26 includes examples on building web application GUIs, validating forms and session tracking. Chapter 27 discusses data-driven and Ajax-enabled JSF applications. The chapter features a database-driven multitier web address book that allows users to add and search for contacts.
- *Web services.* Chapter 28, Web Services, demonstrates creating and consuming SOAP- and REST-based web services. Case studies include developing blackjack and airline reservation web services.
- *Java Web Start and the Java Network Launch Protocol (JNLP).* We introduce Java Web Start and JNLP, which enable applets *and* applications to be launched via a web browser. Users can install locally for later execution. Programs can also request the user’s permission to access local system resources such as files—en-

abling you to develop more robust applets and applications that execute safely using Java's sandbox security model, which applies to downloaded code.

### *Multithreading*

- *Multithreading.* We completely reworked Chapter 23, Multithreading [special thanks to the guidance of Brian Goetz and Joseph Bowbeer—two of the co-authors of *Java Concurrency in Practice*, Addison-Wesley, 2006].
- *SwingWorker class.* We use class `SwingWorker` to create *multithreaded user interfaces*.

### *GUI and Graphics*

- *GUI and graphics presentation.* Chapters 14, 15 and 22, and Appendix H present Java GUI and Graphics programming.
- *GroupLayout layout manager.* We discuss the `GroupLayout` layout manager in the context of the GUI design tool in the NetBeans IDE.
- *JTable sorting and filtering capabilities.* Chapter 25 uses these capabilities to sort the data in a `JTable` and filter it by regular expressions.

### *Other Features*

- *Android.* Because of the tremendous interest in Android-based smartphones and tablets, we've included a three-chapter introduction to Android app development online at [www.deitel.com/books/javafp](http://www.deitel.com/books/javafp). These chapters are from our new Deitel Developer Series book *Android for Programmers: An App-Driven Approach*. After you learn Java, you'll find it straightforward to develop and run Android apps on the free Android emulator that you can download from [developer.android.com](http://developer.android.com).
- *Software engineering community concepts.* We discuss agile software development, refactoring, design patterns, LAMP, SaaS (Software as a Service), PaaS (Platform as a Service), cloud computing, open-source software and more.

## Teaching Approach

*Java for Programmers, 2/e*, contains hundreds of complete working examples. We stress program clarity and concentrate on building well-engineered software.

*Syntax Shading.* For readability, we syntax shade the code, similar to the way most integrated-development environments and code editors syntax color the code. Our syntax-shading conventions are:

```

comments appear like this
keywords appear like this
constants and literal values appear like this
all other code appears in black

```

*Code Highlighting.* We place gray rectangles around each program's key code.

*Using Fonts for Emphasis.* We place the key terms and the index's page reference for each defining occurrence in **bold** text for easier reference. On-screen components are emphasized in the **bold Helvetica** font (e.g., the **File** menu) and Java program text in the **Lucida** font (e.g., `int x = 5;`).



- [download The Makers of Scotland: Picts, Romans, Gaels and Vikings](#)
- [read The Best Little Boy in the World](#)
- [read online Do It Yourself Psychic Power: Practical Tools and Techniques for Awakening Your Natural Gifts using Clairvoyance, Spirit Guides, Chakra Healing, Space Clearing and Aura Reading here](#)
- [download online How to Change Your Thinking About Anxiety: A Hazelden Quick Guide here](#)
- [read online Baseball in Long Beach \(Images of Baseball\) pdf](#)
  
- <http://tuscalaural.com/library/Cooking-with-Vegetables--Crisp--Fresh-and-Full-of-Vitamins.pdf>
- <http://www.gruppoacusma.com/?freebooks/The-Kagonesti--Dragonlance--Lost-Histories--Book-1-.pdf>
- <http://metromekanik.com/ebooks/Do-It-Yourself-Psychic-Power--Practical-Tools-and-Techniques-for-Awakening-Your-Natural-Gifts-using-Clairvoyance>
- <http://hasanetmekci.com/ebooks/Cracking-the-New-SAT--Premium-Edition--2016-Edition-.pdf>
- <http://www.khoi.dk/?books/Baseball-in-Long-Beach--Images-of-Baseball-.pdf>