

GLOBAL  
EDITION



# Android™

## *How to Program*

SECOND EDITION



Paul Deitel • Harvey Deitel • Abbey Deitel

ALWAYS LEARNING

PEARSON

# ONLINE ACCESS

Thank you for purchasing a new copy of *Android™ How to Program, Second Edition*. Your textbook includes 12 months of prepaid access to the book's Companion Website. This prepaid subscription provides you with full access to the following student support areas:

- Source code
- Premium web chapters

**Use a coin to scratch off the coating and reveal your student access code.  
Do not use a knife or other sharp object as it may damage the code.**

To access the *Android How to Program, Second Edition*, Companion Website for the first time, you will need to register online using a computer with an Internet connection and a web browser. The process takes just a couple of minutes and only needs to be completed once.

- 1.** Go to [www.pearsonglobaleditions.com/deitel](http://www.pearsonglobaleditions.com/deitel)
- 2.** Click on **Companion Website**.
- 3.** Click on the **Register** button.
- 4.** On the registration page, enter your student access code\* found beneath the scratch-off panel. Do not type the dashes. You can use lower- or uppercase.
- 5.** Follow the on-screen instructions. If you need help at any time during the online registration process, simply click the **Need Help?** icon.
- 6.** Once your personal Login Name and Password are confirmed, you can begin using the *Android How to Program* Companion Website!

## **To log in after you have registered:**

You only need to register for this Companion Website once. After that, you can log in any time at [www.pearsonglobaleditions.com/deitel](http://www.pearsonglobaleditions.com/deitel) by providing your Login Name and Password when prompted.

\*Important: The access code can only be used once. This subscription is valid for 12 months upon activation and is not transferable. If this access code has already been revealed, it may no longer be valid. If this is the case, you can purchase a subscription by going to [www.pearsonglobaleditions.com/deitel](http://www.pearsonglobaleditions.com/deitel) going to the Android book and following the on-screen instructions.

# Android™

## HOW TO PROGRAM

SECOND EDITION  
Global Edition

# Deitel® Series Page

---

## How To Program Series

Android How to Program, 2/e  
C++ How to Program, 9/E  
C How to Program, 7/E  
Java™ How to Program, 10/E  
Java™ How to Program, Late Objects Version, 10/E  
Internet & World Wide Web How to Program, 5/E  
Visual C++® 2008 How to Program, 2/E  
Visual Basic® 2012 How to Program, 6/E  
Visual C#® 2012 How to Program, 5/E

---

## Simply Series

Simply C++: An App-Driven Tutorial Approach  
Simply Java™ Programming: An App-Driven Tutorial Approach  
Simply C#: An App-Driven Tutorial Approach  
Simply Visual Basic® 2010: An App-Driven Approach, 4/E

---

## CourseSmart Web Books

[www.deitel.com/books/CourseSmart/](http://www.deitel.com/books/CourseSmart/)  
C++ How to Program, 8/E and 9/E  
Simply C++: An App-Driven Tutorial Approach  
Java™ How to Program, 9/E and 10/E  
Simply Visual Basic 2010: An App-Driven Approach, 4/E

---

(continued from previous column)

Visual Basic® 2012 How to Program, 6/E  
Visual Basic® 2010 How to Program, 5/E  
Visual C#® 2012 How to Program, 5/E  
Visual C#® 2010 How to Program, 4/E

---

## Deitel® Developer Series

Android for Programmers: An App-Driven Approach, 2/e, Volume 1  
C for Programmers with an Introduction to C11  
C++11 for Programmers  
C# 2012 for Programmers  
Dive Into® iOS 6 for Programmers: An App-Driven Approach  
Java™ for Programmers, 2/e  
JavaScript for Programmers

---

## LiveLessons Video Learning Products

[www.deitel.com/books/LiveLessons/](http://www.deitel.com/books/LiveLessons/)  
Android App Development Fundamentals,  
C++ Fundamentals  
Java™ Fundamentals  
C# 2012 Fundamentals  
C# 2010 Fundamentals  
iOS® 6 App Development Fundamentals  
JavaScript Fundamentals  
Visual Basic Fundamentals

---

To receive updates on Deitel publications, Resource Centers, training courses, partner offers and more, please join the Deitel communities on

- Facebook®—[facebook.com/DeitelFan](https://facebook.com/DeitelFan)
- Twitter®—[@deitel](https://twitter.com/deitel)
- Google+™—[google.com/+DeitelFan](https://google.com/+DeitelFan)
- YouTube™—[google.com/+DeitelFan](https://google.com/+DeitelFan)
- LinkedIn®—[linkedin.com/company/deitel-&-associates](https://linkedin.com/company/deitel-&-associates)

and register for the free *Deitel® Buzz Online* e-mail newsletter at:

[www.deitel.com/newsletter/subscribe.html](http://www.deitel.com/newsletter/subscribe.html)

To communicate with the authors, send e-mail to:

[deitel@deitel.com](mailto:deitel@deitel.com)

For information on *Dive-Into® Series* on-site seminars offered by Deitel & Associates, Inc. worldwide, write to us at [deitel@deitel.com](mailto:deitel@deitel.com) or visit:

[www.deitel.com/training/](http://www.deitel.com/training/)

For continuing updates on Pearson/Deitel publications visit:

[www.deitel.com](http://www.deitel.com)  
[www.pearsonglobaleditions.com/Deitel](http://www.pearsonglobaleditions.com/Deitel)

Visit the Deitel Resource Centers that will help you master programming languages, software development, Android and iOS app development, and Internet- and web-related topics:

[www.deitel.com/ResourceCenters.html](http://www.deitel.com/ResourceCenters.html)

# Android™

## HOW TO PROGRAM

SECOND EDITION  
Global Edition

**Paul Deitel • Harvey Deitel • Abbey Deitel**

*Deitel & Associates, Inc.*

**Global Edition contributions by Muthuraj M.**



**PEARSON**

Boston Columbus Indianapolis New York San Francisco Upper Saddle River  
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montréal Toronto  
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director, ECS: *Marcia Horton*  
Head of Learning Asset Acquisition, Global Edition:  
*Laura Dent*  
Executive Editor: *Tracy Johnson (Dunkelberger)*  
Director of Marketing: *Christy Lesko*  
Marketing Manager: *Yez Alayan*  
Marketing Assistant: *Jon Bryant*  
Director of Program Management: *Erin Gregg*  
Program Management-Team Lead: *Scott Disanno*  
Program Manager: *Carole Snyder*  
Project Management-Team Lead: *Laura Burgess*  
Project Manager: *Robert Engelhardt*  
Publishing Administrator and Business Analyst,  
Global Edition: *Shokhi Shah Khandelwal*  
Acquisitions Editor, Global Edition: *Karthik Subramanian*

Assistant Project Editor, Global Edition: *Sinjita Basu*  
Media Producer, Global Edition: *M. Vikram Kumar*  
Senior Manufacturing Controller, Production, Global  
Edition: *Trudy Kimber*  
Procurement Specialist: *Linda Sager*  
Permissions Supervisor: *Michael Joyce*  
Permissions Administrator: *Jenell Forschler*  
Director, Image Asset Services: *Annie Atherton*  
Manager, Visual Research: *Karen Sanatar*  
Media Project Manager: *Renata Butera*  
Cover Designer: *Shree Inbakumar*  
Cover Photo: *Kirill\_M/ Shutterstock*  
Cover Printer: *Courier Westford*

---

Pearson Education Limited  
Edinburgh Gate  
Harlow  
Essex CM20 2JE  
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:  
[www.pearsonglobaleditions.com](http://www.pearsonglobaleditions.com)

© Pearson Education Limited 2015

The rights of Paul Deitel, Harvey Deitel, and Abbey Deitel to be identified as the authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

*Authorized adaptation from the United States edition, entitled Android: How to Program, 2nd edition, ISBN 978-0-13-376403-1, by Paul Deitel, Harvey Deitel, and Abbey Deitel, published by Pearson Education © 2015.*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without either the prior written permission of the publisher or a license permitting restricted copying in the United Kingdom issued by the Copyright Licensing Agency Ltd, Saffron House, 6-10 Kirby Street, London EC1N 8TS.

All trademarks used herein are the property of their respective owners. The use of any trademark in this text does not vest in the author or publisher any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of this book by such owners.

ISBN 10: 0-273-79339-X

ISBN 13: 978-0-273-79339-7

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

14 13 12 11 10

Typeset in Adobe Garamond by GEX Publishing Services.

Printed and bound by Courier Westford in the United States of America.

*In Memory of Amar G. Bose, MIT Professor and  
Founder and Chairman of the Bose Corporation:*

*It was a privilege being your student—and members  
of the next generation of Deitels, who heard our dad  
say how your classes inspired him to do his best work.  
You taught us that if we go after the really hard prob-  
lems, then great things can happen.*

*Harvey Deitel  
Paul and Abbey Deitel*

## Trademarks

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

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

Google, Android, Google Play, Google Maps, Google Wallet, Nexus, YouTube, AdSense and AdMob are trademarks of Google, Inc.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. Screen shots and icons reprinted with permission from the Microsoft Corporation. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

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.



<b>Preface</b>	<b>19</b>
<b>Before You Begin</b>	<b>31</b>
<b>I Introduction to Android</b>	<b>39</b>
1.1 Introduction	40
1.2 Android—The World’s Leading Mobile Operating System	41
1.3 Android Features	41
1.4 Android Operating System	45
1.4.1 Android 2.2 (Froyo)	45
1.4.2 Android 2.3 (Gingerbread)	46
1.4.3 Android 3.0 through 3.2 (Honeycomb)	46
1.4.4 Android 4.0 through 4.0.4 (Ice Cream Sandwich)	46
1.4.5 Android 4.1–4.3 (Jelly Bean)	47
1.4.6 Android 4.4 (KitKat)	48
1.5 Downloading Apps from Google Play	49
1.6 Packages	50
1.7 Android Software Development Kit (SDK)	51
1.8 Object-Oriented Programming: A Quick Refresher	54
1.8.1 The Automobile as an Object	55
1.8.2 Methods and Classes	55
1.8.3 Instantiation	55
1.8.4 Reuse	55
1.8.5 Messages and Method Calls	55
1.8.6 Attributes and Instance Variables	56
1.8.7 Encapsulation	56
1.8.8 Inheritance	56
1.8.9 Object-Oriented Analysis and Design (OOAD)	56
1.9 Test-Driving the <b>Doodlz</b> App in an Android Virtual Device (AVD)	57
1.9.1 Running the <b>Doodlz</b> App in the Nexus 4 Smartphone AVD	57
1.9.2 Running the <b>Doodlz</b> App in a Tablet AVD	66
1.9.3 Running the <b>Doodlz</b> App on an Android Device	68
1.10 Building Great Android Apps	68
1.11 Android Development Resources	70
1.12 Wrap-Up	73

## 2 Welcome App 76

*Dive-Into® the Android Developer Tools: Introducing Visual GUI Design, Layouts, Accessibility and Internationalization*

2.1	Introduction	77
2.2	Technologies Overview	78
2.2.1	Android Developer Tools IDE	78
2.2.2	TextViews and ImageViews	78
2.2.3	App Resources	78
2.2.4	Accessibility	78
2.2.5	Internationalization	78
2.3	Creating an App	79
2.3.1	Launching the Android Developer Tools IDE	79
2.3.2	Creating a New Project	79
2.3.3	<b>New Android Application</b> Dialog	80
2.3.4	<b>Configure Project</b> Step	81
2.3.5	<b>Configure Launcher Icon</b> Step	81
2.3.6	<b>Create Activity</b> Step	83
2.3.7	<b>Blank Activity</b> Step	84
2.4	Android Developer Tools Window	85
2.4.1	<b>Package Explorer</b> Window	86
2.4.2	Editor Windows	86
2.4.3	<b>Outline</b> Window	86
2.4.4	App Resource Files	86
2.4.5	<b>Graphical Layout</b> Editor	87
2.4.6	The Default GUI	87
2.5	Building the App's GUI with the <b>Graphical Layout</b> Editor	89
2.5.1	Adding Images to the Project	89
2.5.2	Changing the <b>Id</b> Property of the <b>RelativeLayout</b> and the <b>TextView</b>	90
2.5.3	Configuring the <b>TextView</b>	91
2.5.4	Adding <b>ImageViews</b> to Display the Images	95
2.6	Running the <b>Welcome</b> App	97
2.7	Making Your App Accessible	98
2.8	Internationalizing Your App	100
2.9	Wrap-Up	104

## 3 Tip Calculator App 107

*Introducing GridLayout, LinearLayout, EditText, SeekBar, Event Handling, NumberFormat and Defining App Functionality with Java*

3.1	Introduction	108
3.2	Test-Driving the <b>Tip Calculator</b> App	109
3.3	Technologies Overview	110
3.3.1	Class Activity	110
3.3.2	Activity Lifecycle Methods	110
3.3.3	Arranging Views with <b>LinearLayout</b> and <b>GridLayout</b>	111

3.3.4	Creating and Customizing the GUI with the <b>Graphical Layout Editor</b> and the <b>Outline</b> and <b>Properties</b> Windows	111
3.3.5	Formatting Numbers as Locale-Specific Currency and Percentage Strings	112
3.3.6	Implementing Interface <code>TextWatcher</code> for Handling <code>EditText</code> Text Changes	112
3.3.7	Implementing Interface <code>OnSeekBarChangeListener</code> for Handling <code>SeekBar</code> Thumb Position Changes	112
3.3.8	<code>AndroidManifest.xml</code>	113
3.4	Building the App's GUI	113
3.4.1	<code>GridLayout</code> Introduction	113
3.4.2	Creating the <code>TipCalculator</code> Project	115
3.4.3	Changing to a <code>GridLayout</code>	115
3.4.4	Adding the <code>TextViews</code> , <code>EditText</code> , <code>SeekBar</code> and <code>LinearLayouts</code>	116
3.4.5	Customizing the Views to Complete the Design	118
3.5	Adding Functionality to the App	122
3.6	<code>AndroidManifest.xml</code>	130
3.7	Wrap-Up	131

## 4 Twitter® Searches App 135

*SharedPreferences, Collections, ImageButton, ListView, ListActivity, ArrayAdapter, Implicit Intents and AlertDialogs*

4.1	Introduction	136
4.2	Test-Driving the App	137
4.2.1	Importing the App and Running It	137
4.2.2	Adding a Favorite Search	138
4.2.3	Viewing Twitter Search Results	139
4.2.4	Editing a Search	140
4.2.5	Sharing a Search	142
4.2.6	Deleting a Search	142
4.2.7	Scrolling Through Saved Searches	143
4.3	Technologies Overview	143
4.3.1	<code>ListView</code>	143
4.3.2	<code>ListActivity</code>	144
4.3.3	Customizing a <code>ListActivity</code> 's Layout	144
4.3.4	<code>ImageButton</code>	144
4.3.5	<code>SharedPreferences</code>	144
4.3.6	Intents for Launching Other Activities	145
4.3.7	<code>AlertDialog</code>	145
4.3.8	<code>AndroidManifest.xml</code>	146
4.4	Building the App's GUI	146
4.4.1	Creating the Project	146
4.4.2	<code>activity_main.xml</code> Overview	147
4.4.3	Adding the <code>GridLayout</code> and Components	148

## 10 Contents

4.4.4	Graphical Layout Editor Toolbar	153
4.4.5	ListView Item's Layout: list_item.xml	154
4.5	Building the MainActivity Class	155
4.5.1	package and import Statements	155
4.5.2	Extending ListActivity	157
4.5.3	Fields of Class MainActivity	157
4.5.4	Overriding Activity Method onCreate	158
4.5.5	Anonymous Inner Class That Implements the saveButton's OnClickListener to Save a New or Updated Search	160
4.5.6	addTaggedSearch Method	162
4.5.7	Anonymous Inner Class That Implements the ListView's onItemClickListener to Display Search Results	163
4.5.8	Anonymous Inner Class That Implements the ListView's onItemClickLongClickListener to Share, Edit or Delete a Search	165
4.5.9	shareSearch Method	167
4.5.10	deleteSearch Method	168
4.6	AndroidManifest.xml	170
4.7	Wrap-Up	170

## 5 Flag Quiz App

174

*Fragments, Menus, Preferences, AssetManager, Tweened Animations, Handler, Toasts, Explicit Intents, Layouts for Multiple Device Orientations*

5.1	Introduction	175
5.2	Test-Driving the Flag Quiz App	177
5.2.1	Importing the App and Running It	177
5.2.2	Configuring the Quiz	177
5.2.3	Taking the Quiz	179
5.3	Technologies Overview	181
5.3.1	Menus	181
5.3.2	Fragments	181
5.3.3	Fragment Lifecycle Methods	182
5.3.4	Managing Fragments	182
5.3.5	Preferences	182
5.3.6	assets Folder	182
5.3.7	Resource Folders	183
5.3.8	Supporting Different Screen Sizes and Resolutions	183
5.3.9	Determining the Screen Size	184
5.3.10	Toasts for Displaying Messages	184
5.3.11	Using a Handler to Execute a Runnable in the Future	184
5.3.12	Applying an Animation to a View	184
5.3.13	Logging Exception Messages	185
5.3.14	Using an Explicit Intent to Launch Another Activity in the Same App	185
5.3.15	Java Data Structures	185
5.4	Building the GUI and Resource Files	185

5.4.1	Creating the Project	185
5.4.2	strings.xml and Formatted String Resources	186
5.4.3	arrays.xml	187
5.4.4	colors.xml	188
5.4.5	dimens.xml	188
5.4.6	activity_settings.xml Layout	189
5.4.7	activity_main.xml Layout for Phone and Tablet Portrait Orientation	189
5.4.8	fragment_quiz.xml Layout	189
5.4.9	activity_main.xml Layout for Tablet Landscape Orientation	192
5.4.10	preferences.xml for Specifying the App's Settings	193
5.4.11	Creating the Flag Shake Animation	194
5.5	MainActivity Class	196
5.5.1	package Statement, import Statements and Fields	196
5.5.2	Overridden Activity Method onCreate	197
5.5.3	Overridden Activity Method onStart	199
5.5.4	Overridden Activity Method onCreateOptionsMenu	199
5.5.5	Overridden Activity Method onOptionsItemSelected	200
5.5.6	Anonymous Inner Class That Implements OnSharedPreferenceChangeListener	201
5.6	QuizFragment Class	202
5.6.1	package Statement and import Statements	202
5.6.2	Fields	203
5.6.3	Overridden Fragment Method onCreateView	204
5.6.4	Method updateGuessRows	206
5.6.5	Method updateRegions	207
5.6.6	Method resetQuiz	207
5.6.7	Method loadNextFlag	209
5.6.8	Method getCountryName	211
5.6.9	Anonymous Inner Class That Implements OnClickListener	211
5.6.10	Method disableButtons	214
5.7	SettingsFragment Class	214
5.8	SettingsActivity Class	215
5.9	AndroidManifest.xml	215
5.10	Wrap-Up	216

## 6 Cannon Game App 220

*Listening for Touches, Manual Frame-By-Frame Animation, Graphics, Sound, Threading, SurfaceView and SurfaceHolder*

6.1	Introduction	221
6.2	Test-Driving the Cannon Game App	223
6.3	Technologies Overview	223
6.3.1	Attaching a Custom View to a Layout	223
6.3.2	Using the Resource Folder raw	223
6.3.3	Activity and Fragment Lifecycle Methods	223

## 12 Contents

6.3.4	Overriding View Method onTouchEvent	224
6.3.5	Adding Sound with SoundPool and AudioManager	224
6.3.6	Frame-by-Frame Animation with Threads, SurfaceView and SurfaceHolder	224
6.3.7	Simple Collision Detection	225
6.3.8	Drawing Graphics Using Paint and Canvas	225
6.4	Building the App's GUI and Resource Files	225
6.4.1	Creating the Project	225
6.4.2	strings.xml	226
6.4.3	fragment_game.xml	226
6.4.4	activity_main.xml	227
6.4.5	Adding the Sounds to the App	227
6.5	Class Line Maintains a Line's Endpoints	227
6.6	MainActivity Subclass of Activity	228
6.7	CannonGameFragment Subclass of Fragment	228
6.8	CannonView Subclass of View	230
6.8.1	package and import Statements	230
6.8.2	Instance Variables and Constants	231
6.8.3	Constructor	232
6.8.4	Overriding View Method onSizeChanged	234
6.8.5	Method newGame	235
6.8.6	Method updatePositions	236
6.8.7	Method fireCannonball	239
6.8.8	Method alignCannon	240
6.8.9	Method drawGameElements	241
6.8.10	Method showGameOverDialog	243
6.8.11	Methods stopGame and releaseResources	244
6.8.12	Implementing the SurfaceHolder.Callback Methods	245
6.8.13	Overriding View Method onTouchEvent	246
6.8.14	CannonThread: Using a Thread to Create a Game Loop	247
6.9	Wrap-Up	248

## 7 Doodlz App

253

*Two-Dimensional Graphics, Canvas, Bitmap, Accelerometer, SensorManager, Multitouch Events, MediaStore, Printing, Immersive Mode*

7.1	Introduction	254
7.2	Technologies Overview	256
7.2.1	Using SensorManager to Listen for Accelerometer Events	256
7.2.2	Custom DialogFragments	256
7.2.3	Drawing with Canvas and Bitmap	257
7.2.4	Processing Multiple Touch Events and Storing Lines in Paths	257
7.2.5	Android 4.4 Immersive Mode	257
7.2.6	GestureDetector and SimpleOnGestureListener	257
7.2.7	Saving the Drawing to the Device's Gallery	257

7.2.8	Android 4.4 Printing and the Android Support Library's PrintHelper Class	258
7.3	Building the App's GUI and Resource Files	258
7.3.1	Creating the Project	258
7.3.2	strings.xml	258
7.3.3	dimens.xml	259
7.3.4	Menu for the DoodleFragment	260
7.3.5	activity_main.xml Layout for MainActivity	261
7.3.6	fragment_doodle.xml Layout for DoodleFragment	261
7.3.7	fragment_color.xml Layout for ColorDialogFragment	262
7.3.8	fragment_line_width.xml Layout for LineWidthDialogFragment	264
7.3.9	Adding Class EraseImageDialogFragment	265
7.4	MainActivity Class	266
7.5	DoodleFragment Class	267
7.6	DoodleView Class	274
7.7	ColorDialogFragment Class	286
7.8	LineWidthDialogFragment Class	289
7.9	EraseImageDialogFragment Class	293
7.10	Wrap-Up	294

## **8 Address Book App** **298**

*ListFragment, FragmentTransactions and the Fragment Back Stack, Threading and AsyncTasks, CursorAdapter, SQLite and GUI Styles*

8.1	Introduction	299
8.2	Test-Driving the Address Book App	302
8.3	Technologies Overview	302
8.3.1	Displaying Fragments with FragmentTransactions	303
8.3.2	Communicating Data Between a Fragment and a Host Activity	303
8.3.3	Method onSaveInstanceState	303
8.3.4	Defining Styles and Applying Them to GUI Components	303
8.3.5	Specifying a Background for a TextView	303
8.3.6	Extending Class ListFragment to Create a Fragment That Contains a ListView	304
8.3.7	Manipulating a SQLite Database	304
8.3.8	Performing Database Operations Outside the GUI Thread with AsyncTasks	304
8.4	Building the GUI and Resource Files	304
8.4.1	Creating the Project	304
8.4.2	Creating the App's Classes	305
8.4.3	strings.xml	305
8.4.4	styles.xml	306
8.4.5	textview_border.xml	307
8.4.6	MainActivity's Layout: activity_main.xml	308
8.4.7	DetailsFragment's Layout: fragment_details.xml	308
8.4.8	AddEditFragment's Layout: fragment_add_edit.xml	310
8.4.9	Defining the Fragments' Menus	311

## **14** Contents

8.5	MainActivity Class	312
8.6	ContactListFragment Class	318
8.7	AddEditFragment Class	325
8.8	DetailsFragment Class	331
8.9	DatabaseConnector Utility Class	339
8.10	Wrap-Up	344

## **9** Google Play and App Business Issues **348**

9.1	Introduction	349
9.2	Preparing Your Apps for Publication	349
9.2.1	Testing Your App	350
9.2.2	End User License Agreement	350
9.2.3	Icons and Labels	350
9.2.4	Versioning Your App	351
9.2.5	Licensing to Control Access to Paid Apps	351
9.2.6	Obfuscating Your Code	351
9.2.7	Getting a Private Key for Digitally Signing Your App	352
9.2.8	Screenshots	352
9.2.9	Promotional App Video	353
9.3	Pricing Your App: Free or Fee	354
9.3.1	Paid Apps	355
9.3.2	Free Apps	355
9.4	Monetizing Apps with In-App Advertising	356
9.5	Monetizing Apps: Using In-App Billing to Sell Virtual Goods	357
9.6	Registering at Google Play	358
9.7	Setting Up a Google Wallet Merchant Account	359
9.8	Uploading Your Apps to Google Play	360
9.9	Launching the <b>Play Store</b> from Within Your App	361
9.10	Managing Your Apps in Google Play	362
9.11	Other Android App Marketplaces	362
9.12	Other Popular Mobile App Platforms	362
9.13	Marketing Your Apps	363
9.14	Wrap-Up	367

## **A** Introduction to Java Applications **370**

A.1	Introduction	371
A.2	Your First Program in Java: Printing a Line of Text	371
A.3	Modifying Your First Java Program	375
A.4	Displaying Text with <code>printf</code>	377
A.5	Another Application: Adding Integers	377
A.6	Memory Concepts	381
A.7	Arithmetic	382
A.8	Decision Making: Equality and Relational Operators	385
A.9	Wrap-Up	389



<b>B</b>	<b>Introduction to Classes, Objects, Methods and Strings</b>	<b>394</b>
B.1	Introduction	395
B.2	Declaring a Class with a Method and Instantiating an Object of a Class	395
B.3	Declaring a Method with a Parameter	398
B.4	Instance Variables, <i>set</i> Methods and <i>get</i> Methods	401
B.5	Primitive Types vs. Reference Types	405
B.6	Initializing Objects with Constructors	406
B.7	Floating-Point Numbers and Type <code>double</code>	408
B.8	Wrap-Up	412
<b>C</b>	<b>Control Statements</b>	<b>416</b>
C.1	Introduction	417
C.2	Algorithms	417
C.3	Pseudocode	418
C.4	Control Structures	418
C.5	<code>if</code> Single-Selection Statement	419
C.6	<code>if...else</code> Double-Selection Statement	419
C.7	<code>while</code> Repetition Statement	422
C.8	Case Study: Counter-Controlled Repetition	422
C.9	Case Study: Sentinel-Controlled Repetition	426
C.10	Case Study: Nested Control Statements	431
C.11	Compound Assignment Operators	434
C.12	Increment and Decrement Operators	434
C.13	Primitive Types	436
C.14	Essentials of Counter-Controlled Repetition	437
C.15	<code>for</code> Repetition Statement	438
C.16	Examples Using the <code>for</code> Statement	440
C.17	<code>do...while</code> Repetition Statement	442
C.18	<code>switch</code> Multiple-Selection Statement	443
C.19	<code>break</code> and <code>continue</code> Statements	450
C.20	Logical Operators	450
C.21	Wrap-Up	453
<b>D</b>	<b>Methods: A Deeper Look</b>	<b>461</b>
D.1	Introduction	462
D.2	Program Modules in Java	462
D.3	<code>static</code> Methods, <code>static</code> Fields and Class <code>Math</code>	463
D.4	Declaring Methods with Multiple Parameters	465
D.5	Notes on Declaring and Using Methods	468
D.6	Method-Call Stack and Activation Records	469
D.7	Argument Promotion and Casting	469
D.8	Java API Packages	470

D.9	Introduction to Random-Number Generation	471
	D.9.1 Scaling and Shifting of Random Numbers	472
	D.9.2 Random-Number Repeatability for Testing and Debugging	473
D.10	Case Study: A Game of Chance; Introducing Enumerations	474
D.11	Scope of Declarations	478
D.12	Method Overloading	480
D.13	Wrap-Up	483

## **E Arrays and ArrayLists** **490**

E.1	Introduction	491
E.2	Arrays	491
E.3	Declaring and Creating Arrays	492
E.4	Examples Using Arrays	493
E.5	Case Study: Card Shuffling and Dealing Simulation	502
E.6	Enhanced for Statement	506
E.7	Passing Arrays to Methods	507
E.8	Case Study: Class GradeBook Using an Array to Store Grades	511
E.9	Multidimensional Arrays	516
E.10	Case Study: Class GradeBook Using a Two-Dimensional Array	520
E.11	Class Arrays	526
E.12	Introduction to Collections and Class ArrayList	528
E.13	Wrap-Up	531

## **F Classes and Objects: A Deeper Look** **536**

F.1	Introduction	537
F.2	Time Class Case Study	537
F.3	Controlling Access to Members	541
F.4	Referring to the Current Object's Members with the <code>this</code> Reference	542
F.5	Time Class Case Study: Overloaded Constructors	544
F.6	Default and No-Argument Constructors	550
F.7	Composition	551
F.8	Enumerations	554
F.9	Garbage Collection	556
F.10	<code>static</code> Class Members	557
F.11	<code>final</code> Instance Variables	561
F.12	Packages	561
F.13	Package Access	562
F.14	Wrap-Up	562

## **G Object-Oriented Programming: Inheritance and Polymorphism** **565**

G.1	Introduction to Inheritance	566
G.2	Superclasses and Subclasses	567
G.3	<code>protected</code> Members	568

G.4	Relationship between Superclasses and Subclasses	569
G.4.1	Creating and Using a <code>CommissionEmployee</code> Class	569
G.4.2	Creating and Using a <code>BasePlusCommissionEmployee</code> Class	574
G.4.3	Creating a <code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy	579
G.4.4	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>protected</code> Instance Variables	582
G.4.5	<code>CommissionEmployee–BasePlusCommissionEmployee</code> Inheritance Hierarchy Using <code>private</code> Instance Variables	585
G.5	Class Object	590
G.6	Introduction to Polymorphism	591
G.7	Polymorphism: An Example	592
G.8	Demonstrating Polymorphic Behavior	593
G.9	Abstract Classes and Methods	596
G.10	Case Study: Payroll System Using Polymorphism	597
G.10.1	Abstract Superclass <code>Employee</code>	598
G.10.2	Concrete Subclass <code>SalariedEmployee</code>	601
G.10.3	Concrete Subclass <code>HourlyEmployee</code>	603
G.10.4	Concrete Subclass <code>CommissionEmployee</code>	604
G.10.5	Indirect Concrete Subclass <code>BasePlusCommissionEmployee</code>	606
G.10.6	Polymorphic Processing, Operator <code>instanceof</code> and Downcasting	607
G.10.7	Summary of the Allowed Assignments Between Superclass and Subclass Variables	612
G.11	<code>final</code> Methods and Classes	613
G.12	Case Study: Creating and Using Interfaces	614
G.12.1	Developing a Payable Hierarchy	615
G.12.2	Interface <code>Payable</code>	616
G.12.3	Class <code>Invoice</code>	617
G.12.4	Modifying Class <code>Employee</code> to Implement Interface <code>Payable</code>	619
G.12.5	Modifying Class <code>SalariedEmployee</code> for Use in the Payable Hierarchy	621
G.12.6	Using Interface <code>Payable</code> to Process Invoices and Employees Polymorphically	623
G.13	Common Interfaces of the Java API	624
G.14	Wrap-Up	625

## **H Exception Handling: A Deeper Look 629**

H.1	Introduction	630
H.2	Example: Divide by Zero without Exception Handling	630
H.3	Example: Handling <code>ArithmeticExceptions</code> and <code>InputMismatchExceptions</code>	632
H.4	When to Use Exception Handling	637
H.5	Java Exception Hierarchy	637
H.6	<code>finally</code> Block	640
H.7	Stack Unwinding and Obtaining Information from an Exception Object	644
H.8	Wrap-Up	647

<b>I</b>	<b>GUI Components and Event Handling</b>	<b>650</b>
I.1	Introduction	651
I.2	Nimbus Look-and-Feel	651
I.3	Text Fields and an Introduction to Event Handling with Nested Classes	652
I.4	Common GUI Event Types and Listener Interfaces	658
I.5	How Event Handling Works	659
I.6	JButton	660
I.7	JComboBox; Using an Anonymous Inner Class for Event Handling	665
I.8	Adapter Classes	668
I.9	Wrap-Up	669
<b>J</b>	<b>Other Java Topics</b>	<b>670</b>
J.1	Introduction	671
J.2	Collections Overview	671
J.3	Type-Wrapper Classes for Primitive Types	672
J.4	Interface Collection and Class Collections	672
J.5	Lists	673
	J.5.1 ArrayList and Iterator	673
	J.5.2 LinkedList	675
	J.5.3 Views into Collections and Arrays Method asList	678
J.6	Collections Methods	680
	J.6.1 Method sort	680
	J.6.2 Method shuffle	682
J.7	Interface Queue	683
J.8	Sets	683
J.9	Maps	684
J.10	Introduction to Files and Streams	687
J.11	Class File	688
J.12	Introduction to Object Serialization	689
J.13	Introduction to Multithreading	690
J.14	Creating and Executing Threads with the Executor Framework	691
J.15	Overview of Thread Synchronization	695
J.16	Concurrent Collections Overview	696
J.17	Multithreading with GUI	696
J.18	Wrap-Up	703
<b>K</b>	<b>Operator Precedence Chart</b>	<b>706</b>
<b>L</b>	<b>Primitive Types</b>	<b>708</b>
	<b>Index</b>	<b>709</b>

# Preface

*Build a better mousetrap, and the world will beat a path to your door.*

—Ralph Waldo Emerson

Science and technology and the various forms of art,  
all unite humanity in a single and interconnected system.

—Zhores Aleksandrovich Medvede

Welcome to the dynamic world of Android smartphone and tablet app development with the Android Software Development Kit (SDK), the Java™ programming language, the Android Development Tools IDE, and the new and rapidly evolving Android Studio. We present leading-edge mobile computing technologies for students, instructors and professional software developers.

## ***Android How to Program, 2/e***

With this unique book—the second edition of the world’s first Android computer science textbook—you can learn Android even if you don’t know Java and even if you’re a programming novice. This book includes a complete, 300-page introduction to the Java core programming concepts that you’ll need when developing Android apps. The Java content is appropriate for programming novices.

*Android How to Program, 2/e* was formed by merging

- our professional book *Android for Programmers: An App-Driven Approach, 2/e, Volume 1*
- additional online chapters selected from *Android for Programmers: An App-Driven Approach, 2/e, Volume 2*
- condensed, introductory core content on object-oriented Java programming from our college textbook *Java How to Program, 9/e*
- hundreds of Android short-answer questions and app-development exercises we created for this book—most are in the book and many of the short-answer questions are in the test-item file for instructors.

We scoured the Android material, especially the fully coded Android apps, and enumerated the Java features that you’ll need to build these and similar apps. Then we extracted the corresponding Java content from *Java How to Program, 9/e*. That’s a 1500-page book, so it was challenging to whittle down that much content and keep it friendly, even for programming novices.

When you study the Android content, you’ll be thinking like a developer from the start. You’re going to study and build lots of real stuff and you’ll face the kinds of challenges professional developers must deal with. We’ll point you to the online documenta-

tion and forums where you can find additional information and get answers to your questions. We'll also encourage you to read, modify and enhance open-source code as part of your learning process.

## Intended Audiences

There are several audiences for this book. Most commonly, it will be used in upper-level elective college courses and industry professional courses for people familiar with object-oriented programming but who may or may not know Java and want to learn Android app development.

Uniquely, the book can also be used in introductory courses like CS1, intended for programming novices. We recommend that schools typically offering many sections of CS1 in Java consider designating one or two sections for ambitious students who have at least some prior programming experience and who want to work hard to learn a good amount of Java and Android in an aggressively paced one-semester course. The schools may want to list the courses with “honors” or “accelerated” designations. The book works especially well in two-semester introductory programming sequences where the introduction to Java is covered first.

## App-Development Courses

In 2007, Stanford offered a new course called Creating Engaging Facebook Apps. Students worked in teams developing apps, some of which landed in Facebook's top 10, earning some of the student developers millions of dollars.<sup>1</sup> This course gained wide recognition for encouraging student creativity and teamwork. Scores of colleges now offer app-development courses across many social networking and mobile platforms such as Android and iOS. We encourage you to read the online mobile app development syllabi and check out the YouTube™ videos created by instructors and students for many of these courses.

## Android Ecosystem: Competition, Innovation, Explosive Growth and Opportunities

Sales of Android devices and app downloads have been growing exponentially. The first-generation Android phones were released in October 2008. A study by Strategy Analytics showed that by October 2013, Android had 81.3% of the global smartphone market share, compared to 13.4% for Apple, 4.1% for Microsoft and 1% for Blackberry.<sup>2</sup> According to an IDC report, by the end of the first quarter of 2013 Android had 56.5% of the global tablet market share, compared to 39.6% for Apple's iPad and 3.7% for Microsoft Windows tablets.<sup>3</sup>

There are now over one billion Android smartphones and tablets in use,<sup>4</sup> and more than 1.5 million Android devices are being activated daily.<sup>5</sup> According to IDC, Samsung

- 
1. <http://www.businessinsider.com/these-stanford-students-made-millions-taking-a-class-on-facebook-2011-5>.
  2. <http://blogs.strategyanalytics.com/WSS/post/2013/10/31/Android-Captures-Record-81-Percent-Share-of-Global-Smartphone-Shipments-in-Q3-2013.aspx>.
  3. <http://www.idc.com/getdoc.jsp?containerId=prUS24093213>.
  4. <http://www.android.com/kitkat>.
  5. <http://www.technobuffalo.com/2013/04/16/google-daily-android-activations-1-5-million>.

is the leading Android manufacturer, accounting for nearly 40% of Android device shipments in the third quarter of 2013.

Billions of apps have been downloaded from Google Play™—Google’s marketplace for Android Apps. The opportunities for Android app developers are enormous.

Fierce competition among popular mobile platforms and carriers is leading to rapid innovation and falling prices. Competition among the dozens of Android device manufacturers is driving hardware and software innovation within the Android community.

## App-Driven Approach

At the heart of the book is our *app-driven approach*—we present concepts in the context of *seven complete working Android apps* in the print book and more online. We begin each of the app chapters with an *introduction* to the app, an app *test-drive* showing one or more *sample executions*, and a *technologies overview*. We build the app’s GUI and resource files. Then we proceed with a detailed *code walkthrough* of the app’s source code in which we discuss the programming concepts and demonstrate the functionality of the Android APIs used in the app. All the source code is available at the book’s Companion Website [www.pearsonglobaleditions.com/Deitel](http://www.pearsonglobaleditions.com/Deitel). We recommend that you have the source code open in the IDE as you read the book. Figure 1 lists the book’s apps and the key technologies we used to build each.

App	Technologies
Chapter 2, <b>Welcome App</b>	The Android Developer Tools (the Eclipse IDE and the ADT Plugin), visual GUI design, layouts, TextViews, ImageViews, accessibility and internationalization.
Chapter 3, <b>Tip Calculator App</b>	GridLayout, LinearLayout, EditText, SeekBar, event handling, NumberFormat and defining app functionality with Java.
Chapter 4, <b>Twitter® Searches App</b>	SharedPreferences, collections, ImageButton, ListView, ListActivity, ArrayAdapter, implicit intents and AlertDialogs.
Chapter 5, <b>Flag Quiz App</b>	Fragments, menus, preferences, AssetManager, tweened animations, Handler, Toasts, Explicit Intents, layouts for multiple device orientations.
Chapter 6, <b>Cannon Game App</b>	Listening for touches, frame-by-frame animation, graphics, sound, threading, SurfaceView and SurfaceHolder.
Chapter 7, <b>Doodlz App</b>	Two-dimensional graphics, Canvas, Bitmap, accelerometer, SensorManager, multitouch events, MediaStore, printing and Immersive Mode.
Chapter 8, <b>Address Book App</b>	AdapterViews and Adapters

**Fig. 1** | *Android How to Program* apps in the print book.

## Online Chapters and Book Updates

The Companion Website contains additional app-development chapters that introduce property animation, Google Play game services, video, speech synthesis and recognition, GPS, the Maps API, the compass, object serialization, Internet-enabled apps, audio recording and playback, Bluetooth<sup>®</sup>, HTML5 mobile apps and more. **Most of these chapters will be available for fall 2014 courses. For the status of the online chapters and for continuing book updates, visit**

[www.pearsonglobal Editions.com/Deitel](http://www.pearsonglobal Editions.com/Deitel)

Join the Deitel communities on Facebook<sup>®</sup> (<http://www.deitel.com/deitelFan>), Twitter<sup>®</sup> (@deitel), LinkedIn<sup>®</sup> (<http://bit.ly/DeitelLinkedIn>) Google+<sup>™</sup> (<http://google.com/+DeitelFan>), and YouTube<sup>™</sup> (<http://youtube.com/user/DeitelTV>) and subscribe to the *Deitel<sup>®</sup> Buzz Online* newsletter (<http://www.deitel.com/newsletter/subscribe.html>).

## Copyright Notice and Code License

*All of the Android code and Android apps in the book are copyrighted by Deitel & Associates, Inc. The sample Android apps in the book are licensed under a Creative Commons Attribution 3.0 Unported License (<http://creativecommons.org/licenses/by/3.0>), with the exception that they may not be reused in any way in educational tutorials and textbooks, whether in print or digital format. Additionally, the authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or to the documentation contained in this book. The authors and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs. You're welcome to use the apps in the book as shells for your own apps, building on their existing functionality. If you have any questions, contact us at [deitel@deitel.com](mailto:deitel@deitel.com).*

## Getting up to Speed in Java and XML

The Android portion of this book assumes that you already know Java and object-oriented programming. If you're not familiar with these, the appendices provide a condensed, friendly introduction to Java and the object-oriented programming techniques you'll need to develop Android apps. If you're interested in learning Java in more depth, you may want to check out the comprehensive treatment in our textbook *Java How to Program, 10/e* [www.pearsonglobal Editions.com/Deitel](http://www.pearsonglobal Editions.com/Deitel).

Because of the improved Android development tools, we were able to eliminate almost all XML markup in this edition. There are still two small, easy-to-understand XML files you'll need to manipulate. If you're not familiar with XML, see these online tutorials:

- [http://www.deitel.com/articles/xml\\_tutorials/20060401/XMLBasics/](http://www.deitel.com/articles/xml_tutorials/20060401/XMLBasics/)
- [http://www.deitel.com/articles/xml\\_tutorials/20060401/XMLStructuringData/](http://www.deitel.com/articles/xml_tutorials/20060401/XMLStructuringData/)
- <http://www.ibm.com/developerworks/xml/newto/>
- [http://www.w3schools.com/xml/xml\\_what\\_is.asp](http://www.w3schools.com/xml/xml_what_is.asp)



## Key Features of *Android How to Program, 2/e*

- *Android SDK 4.3 and 4.4.* We cover various new Android Software Development Kit (SDK) 4.3 and 4.4 features. [Note: The apps in this book are configured to run on Android devices with Android 4.3 and higher; however, most apps will work in 4.0 and higher by changing their minimum required SDK.]
- *Fragments.* Starting with Chapter 5, we use Fragments to create and manage portions of each app's GUI. You can combine several fragments to create user interfaces that take advantage of tablet screen sizes. You also can easily interchange fragments to make your GUIs more dynamic, as you'll do in Chapter 8.
- *Support for multiple screen sizes and resolutions.* Throughout the app chapters we demonstrate how to use Android's mechanisms for automatically choosing resources (layouts, images, etc.) based on a device's size and orientation.
- *Eclipse-Based Android Development Tools (ADT) coverage in the print book.* The free Android Development Tools (ADT) integrated development environment (IDE)—which includes Eclipse and the ADT plugin—combined with the free Java Development Kit (JDK) provide all the software you'll need to create, run and debug Android apps, export them for distribution (e.g., upload them to Google Play™) and more.
- *Android Studio.* This is the preferred IDE for the future of Android app development. Because this IDE is evolving quickly, we put our discussions of it online at:

[www.pearsonglobaleditions.com/Deitel](http://www.pearsonglobaleditions.com/Deitel)

- *Immersive Mode.* The status bar at the top of the screen and the menu buttons at the bottom can be hidden, allowing your apps to fill more of the screen. Users can access the status bar by swiping down from the top of the screen, and the system bar (with the back button, home button and recent apps button) by swiping up from the bottom.
- *Printing Framework.* Android 4.4 KitKat allows you to add printing functionality to your apps, such as locating available printers over Wi-Fi or the cloud, selecting the paper size and specifying which pages to print.
- *Testing on Android Smartphones, Tablets and the Android Emulator.* For the best app-development experience, you should test your apps on actual Android smartphones and tablets. You can still have a meaningful experience using the Android emulator (see the Before You Begin section), however it's processor-intensive and can be slow—particularly with games that have a lot of moving parts. In Chapter 1, we mention some Android features that are not supported on the emulator.
- *Multimedia.* The apps in the print book use a broad range of Android multimedia capabilities, including graphics, images, frame-by-frame animation and audio. The apps in the online chapters use property animation, video, speech synthesis and speech recognition.
- *Android Best Practices.* We adhere to accepted Android best practices, pointing them out in the detailed code walkthroughs. For more information, visit <http://developer.android.com/guide/practices/index.html>.
- *Java Content in the Appendices Can Be Used With Java SE 6 or Higher.*

- *Java Exception Handling.* We integrate basic exception handling early in the Java content then present a richer treatment in Appendix H; we use exception handling throughout the Android chapters.
- *Classes Arrays and ArrayList; Collections.* Appendix E covers class Arrays—which contains methods for performing common array manipulations—and generic class ArrayList—which implements a dynamically resizable array-like data structure. Appendix J introduces Java’s generic collections that are used frequently in our Android treatment.
- *Java Multithreading.* Maintaining app responsiveness is a key to building robust Android apps and requires extensive use of Android multithreading. Appendix J introduces multithreading fundamentals so that you can understand our use of the Android AsyncTask class in Chapter 8.
- *GUI Presentation.* Appendix I introduces Java GUI development. Android provides its own GUI components, so this appendix presents a few Java GUI components and focuses on nested classes and anonymous inner classes, which are used extensively for event-handling in Android GUIs.

## Working with Open-Source Apps

There are numerous free, open-source Android apps available online which are excellent resources for learning Android app development. We encourage you to download open-source apps and read their source code to understand how they work. Throughout the book you’ll find programming exercises that ask you to modify or enhance existing open-source apps. Our goal is to give you handles on interesting problems that may also inspire you to create new apps using the same technologies. **Caution: The terms of open source licenses vary considerably.** Some allow you to use the app’s source code freely for any purpose, while others stipulate that the code is available for personal use only—not for creating for-sale or publicly available apps. **Be sure to read the licensing agreements carefully.** If you wish to create a commercial app based on an open-source app, you should consider having an intellectual property attorney read the license; be aware that these attorneys charge significant fees.

## Pedagogic Features

*Syntax Shading.* For readability, we syntax shade the code, similar to Eclipse’s and Android Studio’s use of syntax coloring. Our syntax-shading conventions are as follows:

```

comments appear in gray
constants and literal values appear in bold darker gray
keywords appear in bold black
all other code appears in non-bold black

```

*Code Highlighting.* We emphasize the key code segments in each program by enclosing them in light gray rectangles.

*Using Fonts for Emphasis.* We use various font conventions:

- The defining occurrences of key terms appear in **bold** for easy reference.
- On-screen IDE components appear in **bold Helvetica** (e.g., the File menu).

- Program source code appears in Lucida (e.g., `int x = 5;`).

In this book you'll create GUIs using a combination of visual programming (point and click, drag and drop) and writing code.

We use different fonts when we refer to GUI elements in program code versus GUI elements displayed in the IDE:

- When we refer to a GUI component that we create in a program, we place its class name and object name in a Lucida font—e.g., “Button saveContactButton.”
- When we refer to a GUI component that's part of the IDE, we place the component's text in a **bold Helvetica** font and use a plain text font for the component's type—e.g., “the **File** menu” or “the **Run** button.”

*Using the > Character.* We use the > character to indicate selecting a menu item from a menu. For example, we use the notation **File > New** to indicate that you should select the **New** menu item from the **File** menu.

*Source Code.* All of the book's source code is available for download from:

[www.pearsonglobalEditions.com/Deitel](http://www.pearsonglobalEditions.com/Deitel)

*Chapter Objectives.* Each chapter begins with a list of learning objectives.

*Figures.* Hundreds of tables, source code listings and screen shots are included.

*Software Engineering.* We stress program clarity and performance, and concentrate on building well-engineered, object-oriented software.

*Self-Review Exercises and Answers.* Extensive self-review exercises *and* answers are included for self study.

*Exercises with a Current Flair.* We've worked hard to create topical Android app-development exercises. You'll develop apps using a broad array of current technologies. All of the Android programming exercises require the implementation of complete apps. You'll be asked to enhance the existing chapter apps, develop similar apps, use your creativity to develop your own apps that use the chapter technologies and build new apps based on open-source apps available on the Internet (and again, **be sure to read and comply with the open-source code-license terms for each app**). The Android exercises also include short-answer fill-in and true/false questions.

In the Java exercises, you'll be asked to recall important terms and concepts; indicate what code segments do; indicate what's wrong with a portion of code; write Java statements, methods and classes; and write complete Java programs.

*Index.* We include an extensive index for reference. The page number of the defining occurrence of each key term in the book is highlighted in the index in **bold**.

## **Software Used in *Android How to Program, 2/e***

All the software you'll need for this book is available free for download from the Internet. See the Before You Begin section for the download links.

*Documentation.* All the Android and Java documentation you'll need to develop Android apps is available free at <http://developer.android.com> and <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. The documentation for Eclipse is available at [www.eclipse.org/documentation](http://www.eclipse.org/documentation). The documentation for Android Studio is available at <http://developer.android.com/sdk/installing/studio.html>.

## Instructor Resources

The following supplements are available to **qualified college instructors only** through Pearson Education's Instructor Resource Center [www.pearsonglobaleditions.com/Deitel](http://www.pearsonglobaleditions.com/Deitel):

- *PowerPoint® slides* containing all the code and figures in the text.
- *Test Item File* of short-answer questions.
- *Solutions Manual* with solutions to the end-of-chapter **short-answer exercises** for *both* the Java and Android content. For the Java content, solutions are provided for *most* of the programming exercises.

The suggested Android app-development project exercises are *not* typical homework problems. These tend to be *substantial* projects—many of which could require weeks of effort, possibly with students working in teams. *Selected solutions only* are provided for these project exercises—these will be available on the Pearson Instructor's Resource Center (IRC) for fall semester 2014 classes. Contact us at [deitel@deitel.com](mailto:deitel@deitel.com) if you have any questions.

**Please do not write to us requesting access to the Pearson Instructor's Resource Center. Access is restricted to qualified college instructors teaching from the book. Instructors may obtain access *only* through their Pearson representatives.** If you're not a registered faculty member, contact your Pearson representative.

## Before You Begin

For information configuring your computer so that you can develop apps with Java and Android, see the Before You Begin section that follows this Preface.

## Acknowledgments

Thanks to Barbara Deitel for long hours devoted to this project—she created all of our Java and Android Resource Centers, and patiently researched hundreds of technical details.

This book was a cooperative effort between the academic and professional divisions of Pearson. We appreciate the guidance, wisdom and energy of Tracy Johnson, Executive Editor, Computer Science. Tracy and her team handle all of our academic textbooks. Carole Snyder recruited the book's academic reviewers and managed the review process. Bob Engelhardt managed the book's publication. We selected the cover art and Marta Samsel designed the cover.

We also appreciate the efforts and 18-year mentorship of our friend and professional colleague Mark L. Taub, Editor-in-Chief of the Pearson Technology Group. Mark and his team handle all of our professional books and LiveLessons video products. Kim Boe-

digheimer recruited and managed the professional reviewers for the Android content. John Fuller manages the production of all of our Deitel Developer Series books.

We'd like to thank Michael Morgano, a former colleague of ours at Deitel & Associates, Inc., now an Android developer at Imerj™, who co-authored the first editions of this book and our book, *iPhone for Programmers: An App-Driven Approach*. Michael is an extraordinarily talented software developer.

### ***Reviewers of the Content from Android How to Program and Android for Programmers: An App-Driven Approach Recent Editions***

We wish to acknowledge the efforts of our first and second edition reviewers. They scrutinized the text and the code and provided countless suggestions for improving the presentation: Paul Beusterien (Principal, Mobile Developer Solutions), Eric J. Bowden, COO (Safe Driving Systems, LLC), Tony Cantrell (Georgia Northwestern Technical College), Ian G. Clifton (Independent Contractor and Android App Developer, Daniel Galpin (Android Advocate and author of *Intro to Android Application Development*), Jim Hathaway (Application Developer, Kellogg Company), Douglas Jones (Senior Software Engineer, Fullpower Technologies), Charles Lasky (Nagautuck Community College), Enrique Lopez-Manas (Lead Android Architect, Sixt, and Computer Science Teacher at the University of Alcalá in Madrid), Sebastian Nykopp (Chief Architect, Reaktor), Michael Pardo (Android Developer, Mobiata), Ronan “Zero” Schwarz (CIO, OpenIntents), Arijit Sengupta (Wright State University), Donald Smith (Columbia College), Jesus Ubaldo Quevedo-Torrero (University of Wisconsin, Parkside), Dawn Wick (Southwestern Community College) and Frank Xu (Gannon University).

### ***Reviewers of the Content from Java How to Program Recent Editions***

Lance Andersen (Oracle), Soundararajan Angusamy (Sun Microsystems), Joseph Bowbeer (Consultant), William E. Duncan (Louisiana State University), Diana Franklin (University of California, Santa Barbara), Edward F. Gehringer (North Carolina State University), Huiwei Guan (Northshore Community College), Ric Heishman (George Mason University), Dr. Heinz Kabutz (JavaSpecialists.eu), Patty Kraft (San Diego State University), Lawrence Premkumar (Sun Microsystems), Tim Margush (University of Akron), Sue McFarland Metzger (Villanova University), Shyamal Mitra (The University of Texas at Austin), Peter Pilgrim (Consultant), Manjeet Rege, Ph.D. (Rochester Institute of Technology), Manfred Riem (Java Champion, Consultant, Robert Half), Simon Ritter (Oracle), Susan Rodger (Duke University), Amr Sabry (Indiana University), José Antonio González Seco (Parliament of Andalusia), Sang Shin (Sun Microsystems), S. Sivakumar (Astra Infotech Private Limited), Raghavan “Rags” Srinivas (Intuit), Monica Sweat (Georgia Tech), Vinod Varma (Astra Infotech Private Limited) and Alexander Zuev (Sun Microsystems).

As you read the book, we'd sincerely appreciate your comments, criticisms and suggestions for improving the text. Please address all correspondence to:

`deitel@deitel.com`

We'll respond promptly. We really enjoyed writing this book—we hope you enjoy reading it!

*Paul Deitel*  
*Harvey Deitel*  
*Abbey Deitel*

## About the Authors

**Paul Deitel**, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of MIT, where he studied Information Technology. He holds the Java Certified Programmer and Java Certified Developer certifications, and is an Oracle Java Champion. Through Deitel & Associates, Inc., he has delivered hundreds of programming courses worldwide to clients, including Cisco, IBM, Siemens, Sun Microsystems, Dell, Fidelity, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, White Sands Missile Range, Rogue Wave Software, Boeing, SunGard Higher Education, Nortel Networks, Puma, iRobot, Invensys and many more. He and his co-author, Dr. Harvey M. Deitel, are the world's best-selling programming-language textbook/professional book/video authors.

**Dr. Harvey Deitel**, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 50 years of experience in the computer field. Dr. Deitel earned B.S. and M.S. degrees in Electrical Engineering from MIT and a Ph.D. in Mathematics from Boston University. He has extensive college teaching experience, including earning tenure and serving as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc., in 1991 with his son, Paul Deitel. The Deitels' publications have earned international recognition, with translations published in Simplified Chinese, Traditional Chinese, Korean, Japanese, German, Russian, Spanish, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish. Dr. Deitel has delivered hundreds of programming courses to corporate, academic, government and military clients.

**Abbey Deitel**, President of Deitel & Associates, Inc., is a graduate of Carnegie Mellon University's Tepper School of Management where she received a B.S. in Industrial Management. Abbey has been managing the business operations of Deitel & Associates, Inc. for 16 years. She has contributed to numerous Deitel & Associates publications and, together with Paul and Harvey, is the co-author of *Android for Programmers: An App-Driven Approach*, *2/e*, *iPhone for Programmers: An App-Driven Approach*, *Internet & World Wide Web How to Program*, *5/e*, *Visual Basic 2012 How to Program*, *6/e* and *Simply Visual Basic 2010*, *5/e*.

## Deitel® Dive-Into® Series Programming Languages Training

Deitel & Associates, Inc., founded by Paul Deitel and Harvey Deitel, is an internationally recognized authoring and corporate training organization, specializing in computer programming languages, object technology, mobile app development and Internet and web software technology. The company's training clients include many of the world's largest companies, government agencies, branches of the military, and academic institutions. The company offers instructor-led training courses delivered at client sites worldwide on major programming languages and platforms, including Android app development, Objective-C and iOS app development, Java™, XML®, C++, C, Visual C#®, Visual Basic®, Visual C++®, Python®, object technology, Internet and web programming and a growing list of additional programming and software development courses.

Through its 37-year publishing partnership with Prentice Hall/Pearson, Deitel & Associates, Inc., publishes leading-edge programming college textbooks and professional books in print and a wide range of electronic formats and *LiveLessons* video courses. Deitel & Associates, Inc. and the authors can be reached at:

To learn more about Deitel's *Dive-Into® Series* Corporate Training curriculum, visit:

<http://www.deitel.com/training>

To request a proposal for worldwide on-site, instructor-led training at your organization, e-mail [deitel@deitel.com](mailto:deitel@deitel.com).

Individuals wishing to purchase Deitel books and *LiveLessons* video training can do so through [www.deitel.com](http://www.deitel.com). Bulk orders by corporations, the government, the military and academic institutions should be placed directly with Pearson. For more information, please contact your Pearson representative.

Pearson wishes to thank and acknowledge the following people for their work on the Global Edition:

Contributor:

Muthuraj M., Android Developer

Reviewers:

SC Raghavendra SSE, Intuit, India

Manasa S., NMAM Institute of Technology, Nitte, India

