

CORINNE HOISINGTON'S

ANDROID™ BOOT CAMP

FOR DEVELOPERS USING JAVA™

A GUIDE TO CREATING YOUR FIRST ANDROID APPS

THIRD EDITION



ANDROID™ BOOT CAMP FOR DEVELOPERS USING JAVA™

A GUIDE TO CREATING YOUR FIRST ANDROID APPS

THIRD EDITION

ANDROID™ BOOT CAMP FOR DEVELOPERS USING JAVA™

A GUIDE TO CREATING YOUR FIRST ANDROID APPS

CORINNE HOISINGTON



Australia • Brazil • Mexico • Singapore • United Kingdom • United States

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit www.cengage.com/highered to search by ISBN#, author, title, or keyword for materials in your areas of interest.

Important Notice: Media content referenced within the product description or the product text may not be available in the eBook version.

**Android Boot Camp for Developers Using Java:
A Guide to Creating Your First Android Apps,
Third Edition**

Corinne Hoisington

Product Director: Kathleen McMahon

Product Team Manager: Kristin McNary

Senior Product Manager: Jim Gish

Senior Content Developer: Alyssa Pratt

Development Editor: Lisa Ruffolo,
The Software Resources

Product Assistant: Abigail Pufpaff

Marketing Manager: Eric LaScola

Senior Production Director: Wendy Troeger

Production Director: Patty Stephan

Senior Content Project Manager:
Jennifer K. Feltri-George

Managing Art Director: Jack Pendleton

Cover image(s): ©traffic_analyzer/iStock,
©pictafolio/iStock, ©marty8801/iStock,
©MakaronProduktion/iStock, ©Artos/
Shutterstock

© 2016 Cengage Learning

WCN: 02-200-203

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706

For permission to use material from this text or product, submit all requests online at www.cengage.com/permissions
Further permissions questions can be emailed to
permissionrequest@cengage.com

Library of Congress Control Number: 2015942358

ISBN: 978-1-305-85799-5

Cengage Learning

20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at www.cengage.com

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For your course and learning solutions, visit www.cengage.com

Purchase any of our products at your local college store or at our preferred online store www.cengagebrain.com

Notice to the Reader

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

Printed in the United States of America
Print Number: 01

Print Year: 2016

Brief Contents

	Preface	xvii
CHAPTER 1	Voilà! Meet the Android.	1
CHAPTER 2	Simplify! The Android User Interface	33
CHAPTER 3	Engage! Android User Input, Variables, and Operations	81
CHAPTER 4	Explore! Icons and Decision-Making Controls .	129
CHAPTER 5	Investigate! Android Lists, Arrays, and Web Browsers.	175
CHAPTER 6	Jam! Implementing Audio in Android Apps. . .	219
CHAPTER 7	Reveal! Displaying Pictures in a GridView . . .	261
CHAPTER 8	Design! Using a DatePicker on a Tablet. . . .	299
CHAPTER 9	Customize! Navigating with a Master/ Detail Flow Activity on a Tablet	341
CHAPTER 10	Move! Creating Animation.	379
CHAPTER 11	Discover! Persistent Data.	417
CHAPTER 12	Finale! Publishing Your Android App	455
	Glossary	481
	Index	489

Contents

	Preface	xvii
CHAPTER 1	Voilà! Meet the Android	1
	Meet the Android	2
	Android Phone Device	4
	Features of the Android.	6
	Writing Android Apps	6
	Android Emulator	7
	Getting Oriented with Market Deployment	8
	First Venture into the Android World	9
	Opening Android Studio to Create a New Project	10
	Creating the Hello Android World Project	10
	Building the User Interface	16
	Taking a Tour of the Android Project View	16
	Designing the User Interface Layout within the Virtual Device	17
	Modifying the Text in the TextView Control	19
	Testing the Application in the Emulator.	21
	Opening a Saved App in Android Studio	24
	Wrap It Up—Chapter Summary	25
	Key Terms	26
	Developer FAQs	27
	Beyond the Book	27
	Case Programming Projects	28
	Case Project Famous Technology Quotes App	29
	Case Project Android Dessert Names App.	30
	Case Project Large Tech Companies	31

CHAPTER 2 **Simplify! The Android User Interface 33**

- Designing an Android App 34
 - The Big Picture 35
- Using the Android User Interface 35
 - Relative Layouts and Linear Layouts 42
 - Designing the Healthy Recipes Opening User Interface . . . 44
 - Android Text Properties. 45
 - Adding a File to the Resources Folder 49
 - Adding an ImageView Control 52
 - Adding a Button Control 56
- Creating Activities 57
 - Planning a Program 58
 - Adding a Class File. 58
- The Android Manifest File. 63
- Coding the Java Activity 64
 - Coding an onCreate Method. 64
 - Displaying the User Interface 65
 - Creating a Button Event Handler. 66
 - Coding a Button Event Handler 67
 - Correcting Errors in Code 72
 - Saving and Running the Application 73
- Wrap It Up—Chapter Summary 73
- Key Terms 74
- Developer FAQs 75
- Beyond the Book 76
- Case Programming Projects 76
 - Case Project
 - Youth Hostel App 76
 - Case Project
 - Segway Rental App 78
 - Case Project
 - Your School App 79
 - Case Project
 - Business Card App 79
 - Case Project
 - Your Contacts App—Address Book. 80
 - Case Project
 - Latest Music Scene App 80

CHAPTER 3

Engage! Android User Input, Variables, and Operations	81
Android Themes	83
Previewing a Theme	84
Updating the Theme in the styles.xml File	88
Simplifying User Input	89
Using Android Text Fields	90
Adding a String Array	93
Setting the Hint Property for the Text Field	98
Using the Android Spinner Control	100
Adding the Button, TextView, and ImageView Controls	102
Coding the EditText Class for the Text Field	105
Coding the Spinner Control	107
Instantiating the Button Control	108
Declaring Variables	111
Primitive Data Types	112
String Data Type	113
Declaring the Variables	113
getText() Method	114
Working with Mathematical Operations	116
Arithmetic Operators	116
Formatting Numbers	116
Displaying Android Output	117
getSelectedItem() Method	118
setText() Method	118
Wrap It Up—Chapter Summary	120
Key Terms	121
Developer FAQs	122
Beyond the Book	123
Case Programming Projects	124
Case Project	
Catalina Island Boat Express App	124
Case Project	
Triathlon Registration App	125
Case Project	
Paint Calculator App	126
Case Project	
Chicago Cab Fare App	127
Case Project	
Split the Bill App	127
Case Project	
Piggy Bank Children’s App	128

CHAPTER 4	Explore! Icons and Decision-Making Controls	129
	Using the Launcher Icon	132
	Customizing a Launcher Icon	134
	Displaying the Action Bar Icon Using Code	140
	String Table	141
	RadioButton and RadioGroup Controls	142
	Changing the Text Color of Android Controls	143
	Changing the Margins	144
	Changing the Layout Gravity	144
	Adding the RadioButton Group.	144
	Completing the User Interface	148
	Coding a RadioButton Control	149
	Coding the Button Control	151
	Making Decisions with Conditional Statements	153
	Using an If Statement	153
	Using If Else Statements	154
	Relational Operators	154
	Logical Operators	156
	Data Validation	157
	Toast Notification	157
	Using the isChecked() Method of RadioButton Controls	157
	Using Nested If Statements	158
	Coding the Button Event	158
	Coding the Nested If Statements	160
	Running and Testing the App	164
	Wrap It Up—Chapter Summary	164
	Key Terms	165
	Developer FAQs	166
	Beyond the Book	167
	Case Programming Projects	168
	Case Project	
	Phone Photo Prints App	168
	Case Project	
	Car Wash App	170
	Case Project	
	Power Tool Rental App	172
	Case Project	
	Floor Tiling App	172
	Case Project	
	Currency Conversion App	173
	Case Project	
	Average Income Tax by Country App	174

CHAPTER 5	Investigate! Android Lists, Arrays, and Web Browsers	175
	Creating a List	179
	Extending a ListActivity.	180
	Creating an Array	183
	Declaring an Array	184
	Using a setListAdapter and Array Adapter	185
	Adding the Images to the Resources Folder	187
	Adding the String Table.	188
	Creating a Custom XML Layout for a ListView.	189
	Coding a setListAdapter with a Custom XML Layout	191
	Using the onItemClick Method	193
	Decision Structure—Switch Statement.	195
	Android Intents	197
	Launching the Browser from an Android Device.	198
	Adding Multiple Class Files	200
	Designing XML Layout Files	202
	Opening the Class Files.	205
	Running and Testing the Application	206
	Wrap It Up—Chapter Summary	207
	Key Terms	208
	Developer FAQs	209
	Beyond the Book	210
	Case Programming Projects	211
	Case Project	
	Beach and Mountain Bike Rentals App	211
	Case Project	
	Chocolate Cafe App	214
	Case Project	
	Rent a Car App.	216
	Case Project	
	Coffee Finder App	217
	Case Project	
	Tech Gadgets App	217
	Case Project	
	Create Your Own App	218
CHAPTER 6	Jam! Implementing Audio in Android Apps.	219
	Creating a Splash Screen.	221
	Adding a Background Image to a TextView Widget.	223
	Creating a Timer.	226

- Scheduling a Timer. 229
- Life and Death of an Activity 231
- Launching the Next Activity. 234
- Designing the activity_main.xml File 235
 - Class Variables 238
- Playing Music 242
- Creating a Raw Folder for Music Files 243
- Using the MediaPlayer Class 244
 - The MediaPlayer State 245
 - Changing the Text Property Using Code 248
 - Changing the Visibility Property Using Code 250
- Running and Testing the Application 253
- Wrap It Up—Chapter Summary 253
- Key Terms 254
- Developer FAQs 255
- Beyond the Book 256
- Case Programming Projects 256
 - Case Project
 - Celtic Songs App. 257
 - Case Project
 - Animal Voices Children’s App 258
 - Case Project
 - Serenity Sounds App 259
 - Case Project
 - Guitar Solo App 259
 - Case Project
 - Ring Tones App 260
 - Case Project
 - Your Personal Playlist App. 260

CHAPTER 7 **Reveal! Displaying Pictures in a GridView . . . 261**

- Adding a GridView Control 264
 - Adding the ImageView Control and Image Files 268
- Creating an Array for the Images 271
 - Instantiating the GridView and ImageView Controls 273
- Using a setAdapter with an ImageAdapter 274
- Coding the OnItemClickListener 276
- Coding a Custom Toast Notification 279
- Displaying the Selected Image 281
- Customizing the ImageAdapter Class 282
 - Defining the Context of the ImageAdapter Class 283

Calculating the Length of an Array	284
Coding the getView Method.	286
Running and Testing the Application	290
Wrap It Up—Chapter Summary	290
Key Terms	291
Developer FAQs	292
Beyond the Book	293
Case Programming Projects	293
Case Project	
Quick Healthy Snack Ideas App	294
Case Project	
New Seven Wonders of the World App	295
Case Project	
S.P.C.A. Rescue Shelter App	296
Case Project	
Car Rental App.	297
Case Project	
Anthology Wedding Photography App.	297
Case Project	
Personal Photo App.	298
CHAPTER 8 Design! Using a DatePicker on a Tablet	299
Designing a Tablet Application	302
Design Best Practices for Tablets	303
Adding an Android Virtual Device for the Tablet	304
Creating a Tablet App	304
Creating the String Table	309
Designing a Tablet Table Layout	311
Date, Time, and Clocks	317
Instantiating the Objects	319
Using the Calendar Class.	322
Date Format.	323
DatePickerDialog Input	323
Selecting the Date from the DatePickerDialog	325
Adding the onDateSet() Method	326
Displaying the Date Using the getTime() Method	327
Running and Testing the Application	329
Wrap It Up—Chapter Summary	329
Key Terms	330
Developer FAQs	331
Beyond the Book	332

- Case Programming Projects 332
 - Case Project
 - Appalachian Trail Festival Tablet App 333
 - Case Project
 - The Dog Sledding Experience Tablet App 335
 - Case Project
 - Country Cabin Rental Tablet App 337
 - Case Project
 - Final Touch Auto Detailing Tablet App. 338
 - Case Project
 - Wild Ginger Dinner Delivery Tablet App 339
 - Case Project
 - Create Your Own Tablet App 339

CHAPTER 9

Customize! Navigating with a Master/Detail Flow Activity on a Tablet 341

- Understanding Responsive Design. 345
- Using Application Templates 346
 - Master/Detail Flow Template 347
 - Understanding the Structure of the Master/Detail Flow Template 351
 - Adding Images to the Drawable Folder 353
- Designing an XML TableLayout 354
- Creating a TextView XML Layout for the Second List Item . . . 359
- Creating a WebView XML Layout for the Third List Item . . . 361
 - Customizing the Item List. 363
 - Displaying the Custom Layout in the Detail Pane 367
- Running and Testing the Application 371
- Wrap It Up—Chapter Summary 371
- Key Terms 372
- Developer FAQs 372
- Beyond the Book 373
- Case Programming Projects 374
 - Case Project
 - Oasis Spa Tablet App 374
 - Case Project
 - Modern Art Museums 376
 - Case Project
 - Famous Athlete Tablet App 377
 - Case Project
 - Snap Fitness Tablet App. 377

Case Project	
Top Tablet Apps	378
Case Project	
Pick Your Topic Tablet App	378

CHAPTER 10	Move! Creating Animation	379
	Android Animation	382
	Adding the Layout for the Frame Image and Button Controls	382
	Creating Frame-by-Frame Animation	387
	Coding the AnimationDrawable Object	390
	Setting the Background Resource	391
	Adding Two Button Controls.	394
	Using the Start and Stop Methods	397
	Creating Tween Animation	400
	Creating a Second Activity and XML Layout to Launch the Tween Animation	401
	Adding the Layout for the Tween Image	403
	Coding a Tween Rotation XML File	404
	Coding a StartAnimation	406
	Changing the Emulator to Landscape Orientation	408
	Running and Testing the Application	408
	Wrap It Up—Chapter Summary	408
	Key Terms	410
	Developer FAQs	410
	Beyond the Book	411
	Case Programming Projects	411
	Case Project	
	Facial Expressions App	412
	Case Project	
	Improve Your Golf Stroke App	413
	Case Project	
	Android Rotation App	414
	Case Project	
	Cartoon Animation App	414
	Case Project	
	Flags of the World App	415
	Case Project	
	Frame and Tween Animation Game App	415

CHAPTER 11 Discover! Persistent Data 417

- Understanding Persistent Data 420
 - Using Shared Preferences 420
 - Using Internal Storage 421
 - Using External Storage 421
 - Using SQLite Databases 421
 - Using a Network Connection 422
- Creating XML Layout Files 422
 - Creating a Second Activity and XML Layout. 428
 - Instantiating the XML Controls. 430
 - Writing Persistent Data with SharedPreferences. 433
 - Launching the Second Activity. 436
 - Instantiating the Second Activity Controls 437
- Retrieving Preferences 438
- Coding an ImageView Control 440
- Running and Testing the Application 444
- Wrap It Up—Chapter Summary 444
- Key Terms 445
- Developer FAQs 445
- Beyond the Book 446
- Case Programming Projects 447
 - Case Project
 - BMI Calculator App 447
 - Case Project
 - Home Mortgage Interest App 449
 - Case Project
 - Relocation Moving Truck Rental App 451
 - Case Project
 - Marathon Race App. 452
 - Case Project
 - Amtrak Train App. 453
 - Case Project
 - Your Personal Limerick App 454

CHAPTER 12 Finale! Publishing Your Android App 455

- Understanding Google Play 456
- Targeting Device Configurations and Languages 457
- Adding Localization Using the Translations Editor 458
- Testing Your App on an Android Device 460
- Creating an APK Package 461

Preparing Promotional Materials to Upload466
Providing Images467
Providing a Description467
Including Social Networks470
Registering for a Google Play Account470
Uploading an App to Google Play472
Wrap It Up—Chapter Summary477
Key Terms478
Developer FAQs478
Beyond the Book479
Glossary481
Index489

Preface

Welcome to *Android Boot Camp for Developers Using Java: A Guide to Creating Your First Android Apps, Third Edition*. This book is designed for people who have some programming experience or are new to Java programming and want to move into the exciting world of developing apps for Android mobile devices on a Windows or Mac computer. Google Android is quickly becoming the operating system of choice for mobile devices, including smartphones and tablets, with nearly three-quarters of the world's mobile devices running on the Android platform. To help you participate in the growing Android market, this book focuses on developing apps for Android devices.

Approach

The approach used in *Android Boot Camp for Developers Using Java, Third Edition*, is straightforward. You review a completed Android app and identify why people use the app, and then analyze the tasks it performs and the problems it solves. You also learn about the programming logic, Java tools, and code syntax you can use to create the app. Next, you work through a hands-on tutorial that guides you through the steps of creating the Android app, including designing the user interface and writing the code. After each step, you can compare your work to an illustration that shows exactly how the interface should look or what the code should contain. Using the illustrations, you can avoid mistakes in creating the app and finish the chapter with an appealing, real-world Android app.

The main tool used in *Android Boot Camp for Developers Using Java, Third Edition*, is the standard one developers use to create Android apps: Android Studio, a free, open-source integrated development environment (IDE). Android Studio includes an emulator for testing your apps, so you don't need a smartphone or tablet to run any of the apps covered in this book. Instructions for downloading and setting up Android Studio are provided later in this preface.

What This Book Is

This book introduces you to writing apps for Android mobile devices. It familiarizes you with the development software for creating Android apps, programming logic used in the apps, and Java code that puts the software design and logic into practice. You don't need an Android device because you can run the apps you create in this book by using an Android emulator.

What This Book Is Not

Because this book is targeted to those new to developing Android apps, it doesn't cover advanced topics, such as application programming interfaces (APIs) for each platform. Instead, this book provides a launch pad to begin your journey into creating Android apps for fun and for profit.

xviii

In addition, this book is not an exhaustive information resource. You can find a wealth of information, tutorials, examples, and other resources for the Android platform online. You should learn enough from this book that you can modify and make use of code you find to fit your needs. The best way to learn how to create Android apps is to write code, make mistakes, and learn how to fix them.

Organization and Coverage

Chapter 1 introduces the Android platform and describes the current market for Android apps. You create your first Android project using Android Studio and become familiar with the Android Studio interface and its tools. As programming tradition mandates, your first project is called Hello Android World, which you complete and then run in an emulator.

Chapter 2 focuses on the Android user interface. While developing an app for selecting and displaying healthy recipes, you follow a series of steps that you repeat every time you create an Android app. You learn how to develop a user interface using certain types of controls, select a screen layout, and write code that responds to a button event (such as a tap or click). While creating the chapter project, you develop an app that includes more than one screen and can switch from one screen to another. Finally, you learn how to correct errors in Java code.

Chapter 3 covers user input, variables, and operations. You develop an Android app that allows users to enter the number of concert tickets they want to purchase, and then tap or click a button to calculate the total cost of the tickets. To do so, you create a user interface using an Android theme and add controls to the interface, including text fields, buttons, and spinner controls. You also declare variables and use arithmetic operations to perform calculations, and then convert and format numeric data.

Chapter 4 discusses icons and decision-making controls. The sample app provides healthcare professionals with a mobile way to convert the weight of a patient from pounds to kilograms and from kilograms to pounds. You create this project using a custom application icon, learn how to fine-tune the layout of the user interface, and include radio buttons for user selections. You also learn how to program decisions using If statements, If Else statements, and logical operators.

Chapter 5 describes how to use lists, arrays, and web browsers in an Android app. You design and create an Android app that people can use as a traveler's guide to popular attractions in Chicago, Illinois. To do so, you work with lists, images, and the Switch decision structure. You also learn how to let users access a web browser while using an Android app.

Chapter 6 explains how to include audio such as music in Android apps. The sample app opens with a splash screen and then displays a second screen where users can select a song

to play. To develop this app, you create and set up a splash screen, learn about the Activity life cycle, pause an Activity, and start, play, stop, and resume music playback.

Chapter 7 demonstrates how to use an Android layout tool called a GridView, which shows thumbnail images in a scrolling grid. When the user taps or clicks a thumbnail, the app displays a larger image below the grid. You also learn how to use an array to manage the images.

In **Chapter 8**, you design a calendar program that includes a DatePicker control for selecting a date to book a reservation. Because this app is designed for a larger tablet interface, you also learn how to design an app for a tablet device and add an Android Virtual Device specifically designed for tablets.

Chapter 9 continues to explore Android apps designed for tablet devices. In this chapter, you create a multipane interface, with a list of options in the left pane, and details about the selected option in the right pane. Each pane displays a different layout and Activity. To create the multipane interface, you work with the Master/Detail Flow template.

Chapter 10 explains how to create two types of animation. Using a frame-by-frame animation, you animate a series of images so that they play in sequence. Using a motion tween animation, you apply an animated effect to a single image.

Chapter 11 shows you how to create an Android app that requests data, stores it, and then modifies that data to produce a result throughout multiple activities. You learn about the ways Android apps can save persistent application data, and then use one—the SharedPreferences class—to store data for an airline’s customer rewards app.

In **Chapter 12**, you learn how to publish an Android app to the Google Play Store. Before publishing the app, you test it, prepare it for publication, create a package and digitally sign the app, and then prepare promotional materials.

Features of the Book

Android Boot Camp for Developers Using Java, Third Edition, includes the following features:

- *Objectives*—Each chapter begins with a list of objectives as an overview of the topics discussed in the chapter and as a useful study aid.
- *GTKs, In the Trenches, and Critical Thinking*—GTK stands for Good to Know. These notes offer tips about Android devices, Android apps, and the Android development tools. The In the Trenches features provide programming advice and practical solutions to typical programming problems. The Critical Thinking features pose thought-provoking questions and provide answers related to programming and Java.
- *Figures and tables*—The chapters contain a wealth of screen shots to guide you as you create Android apps and learn about the Android marketplace. In addition, many tables are included to give you an at-a-glance summary of useful information.
- *Step-by-step tutorials*—Starting in Chapter 1, you create complete, working Android apps by performing the steps in a series of hands-on tutorials that lead you through the development process.

- *Code syntax features*—Each new programming concept or technique is introduced with a code syntax feature that highlights a type of statement or programming structure. The code is analyzed and explained thoroughly before you use it in the chapter project.
- *Summaries*—At the end of each chapter is a summary list that recaps the Android terms, programming concepts, and Java coding techniques covered in the chapter so that you have a way to check your understanding of the chapter’s main points.
- *Key terms*—Each chapter includes definitions of new terms, alphabetized for ease of reference. This feature is another useful way to review the chapter’s major concepts.
- *Developer FAQs*—Each chapter contains many short-answer questions that help you review the key concepts in the chapter.
- *Beyond the Book*—In addition to review questions, each chapter provides research topics and questions. You can search the web to find the answers to these questions and further your Android knowledge.
- *Case programming projects*—Except for Chapter 12, each chapter outlines realistic programming projects, including their purpose, algorithms, and conditions. For each project, you use the same steps and techniques you learned in the chapter to create a running Android app on your own.
- *Quality*—Every chapter project and case programming project was tested using Windows 8 and Mac OS X computers.

Student Resources

Source code and project files for the chapter projects and case programming projects in this text are available at www.CengageBrain.com.

For complete instructions on downloading, installing, and setting up the tools you need to perform the steps in this book, see the section titled “Prelude! Installing the Android SDK with Android Studio” later in this preface.

For the Instructor

Android Boot Camp for Developers Using Java, Third Edition, is intended to be taught as a complete course dedicated to the mobile programming of the Android device or as an exploratory topic in a programming class or literacy course. Students can develop Android applications on a Windows or Mac computer using the Android Studio emulator in a traditional or online class. Offering such a stimulating topic that is relative to today’s huge growth in the mobile environment brings excitement to the programming classroom. The Android platform is fully free and open-source, which means all students can access these tools on their home computers.

Instructor Resources

The following teaching tools are available on the Instructor Companion Site (sso.cengage.com) to instructors who have adopted this book:

Instructor's Manual. The Instructor's Manual follows the book chapter by chapter to assist in planning and organizing an effective, engaging course. The manual includes learning objectives, chapter overviews, ideas for classroom activities, and abundant additional resources. A sample course syllabus is also available.

Test Bank. Cengage Learning Testing Powered by Cognero is a flexible, online system that allows you to:

- author, edit, and manage test bank content from multiple Cengage Learning solutions
- create multiple test versions in an instant
- deliver tests from your LMS, your classroom or wherever you want

PowerPoint presentations. PowerPoint slides are available for each chapter. They're offered as a teaching aid for classroom presentations, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics or customize the slides with access to all the figure files from the book.

Solution files. Solution files for all chapter projects and the end-of-chapter exercises are provided.

Prelude! Installing the Android SDK with Android Studio

Setting Up the Android Environment

To begin developing Android applications, you must first set up the Android programming environment on your computer. To establish a development environment, this preface walks you through the installation and setup for a Windows or Mac computer. The Android Software Development Kit (SDK) allows developers to create applications for the Android platform. The Android SDK includes Android Studio along with sample projects including source code, development tools, an emulator, and required libraries to build Android applications, which are written using the Java programming language.

Although installing the Android SDK with Android Studio is easy, you must follow the instructions in this preface to correctly prepare for creating an Android application. Before writing your first application in Chapter 1, complete the following general steps to successfully install the Android SDK with Android Studio on your computer:

1. Prepare your computer for the installation.
2. Download and install the Android Studio Integrated Development Environment (IDE).
3. Add an Android SDK.
4. Set up the Android emulator.

Preparing Your Computer

The Android Software Development Kit is compatible with Windows Vista (32- or 64-bit), Windows 7 (32- or 64-bit), Windows 8.x (32- or 64- or 128-bit), Windows 10 (32- or 64- or 128-bit), Mac OS X (Intel only), and Linux with 2 GB of RAM minimum for all platforms. Java Development Kit (JDK) 7 must already be installed on the PC. To install the basic files needed to write an Android application, your hard drive needs at least 1 GB of available space. Android Studio is the officially recommended IDE of Android app development, replacing Eclipse.

Before getting started, confirm that the latest free Java updates for your computer have been installed. If necessary, go to the site java.com to install the latest Java updates to prepare for the Android installation. You cannot install Android Studio unless the Java updates are installed.

Downloading the Android SDK with Android Studio

The preferred Java program development software for mobile applications is called the Android SDK, which includes Android Studio plus additional tools. The Android SDK provides you with the API libraries and developer tools necessary to build, test, and debug apps for Android. The Android SDK with Android Studio is a free and open-source integrated development environment (IDE). The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs) that run in the Android emulator.

To download the Android SDK with Android Studio and then start Android Studio, complete the following steps. Note that the images shown in the figures may look different on your computer.

STEP 1

- Use a browser to open the webpage <http://developer.android.com/sdk/>.

The Download page opens in the browser (Figure 1).

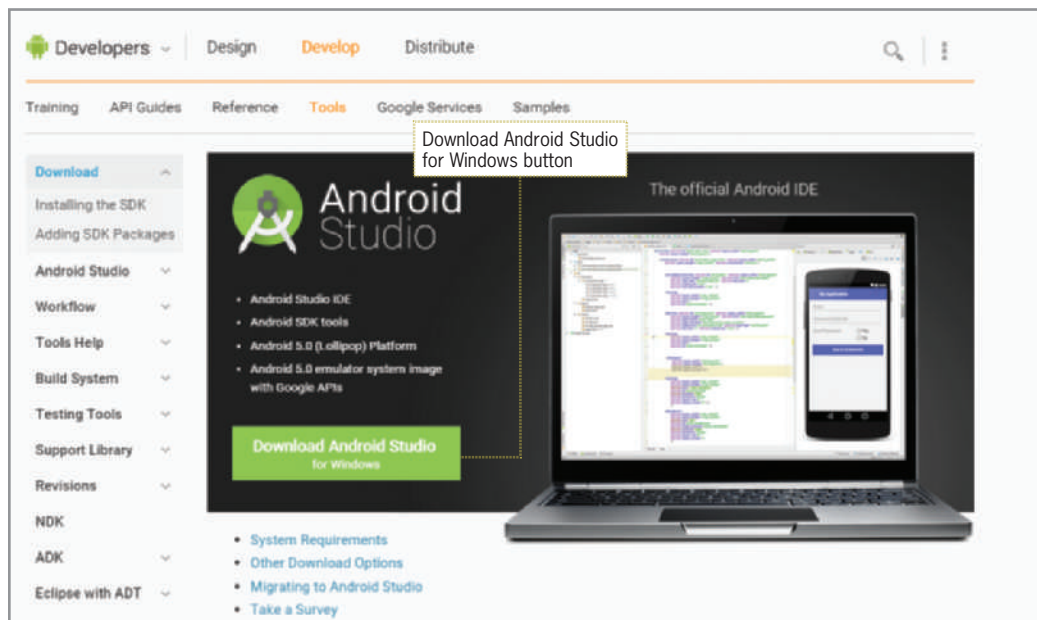


Figure 1 Android Studio download site

STEP 2

- Tap or click the Download Android Studio button to download the most recent version of the Android Studio. *If you are using a Mac, the button is named Download Android Studio for Mac.*

The website detects whether you are installing the Android SDK on a computer running Windows or Mac OS X. The latest Android system image for the emulator is included with this installation.

The Terms and Conditions webpage opens (Figure 2).

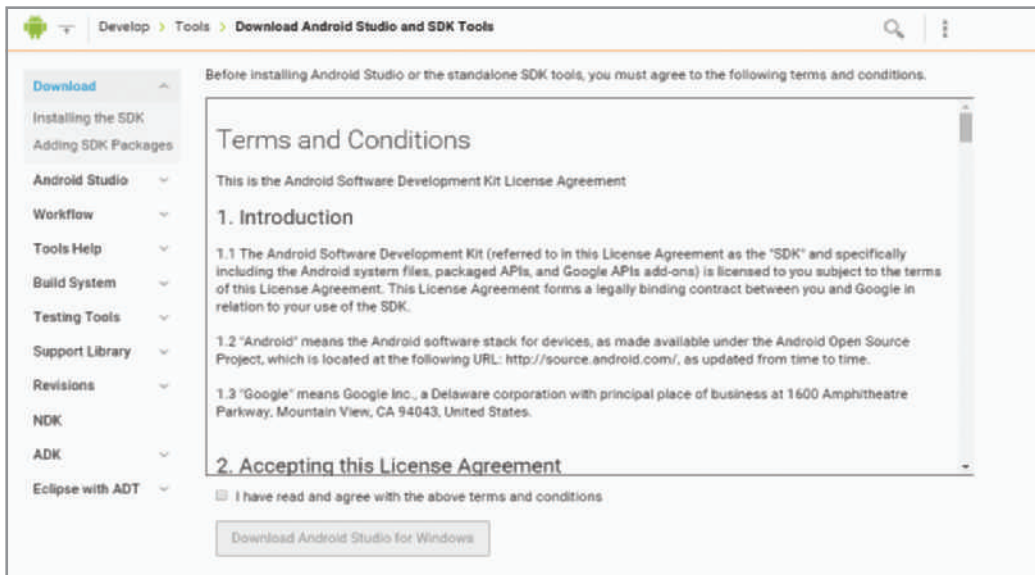


Figure 2 Terms and Conditions of Android Studio

STEP 3

- Scroll down the page and then tap or click the ‘I have read and agree with the above terms and conditions’ check box.
- Tap or click the Download Android Studio for Windows or for Mac button.
- Save the file on your computer.
- Navigate to the location of the downloaded file, and then double-tap or double-click the downloaded file to begin the setup.

The Android Studio Setup dialog box opens (Figure 3).



Figure 3 Android Studio Setup dialog box

STEP 4

- Tap or click the Next button to continue the download and installation process. *If you are using a Mac, drag the Android Studio icon to the Applications folder.*
- Accept the default settings throughout the rest of the download and installation of Android Studio.
- By default, Android Studio opens after the installation of the program.

Android Studio starts. You may want to create a shortcut on the desktop to make it easier to start Android Studio. After successfully downloading the Android Studio, you are ready to begin coding with various pre-installed emulators, ready to test your newly built apps. This textbook uses the Nexus 5 built-in emulator for testing purposes of smartphone apps.

Mac System Requirements

Before you install the software, be sure that your Mac meets the following system requirements by completing the following step:

STEP 1

- Tap or click the Apple icon on the Mac toolbar, and then tap or click About This Mac to open the About This Mac dialog box and view your current operating system version. Your operating system should be Mac OS X 10.5.8 or later (x86 only).
- In the About This Mac dialog box, verify that your Mac has an Intel processor. If the processor is displayed as Unknown, tap or click the More Info button, if possible, to display the processor type.

Acknowledgments

Android Boot Camp for Developers Using Java, A Guide to Creating Your First Android Apps, Third Edition, is the product of a wonderful team of professionals working toward a single goal: providing students with pertinent, engaging material to further their knowledge and careers. Thank you to the folks at Cengage—specifically Senior Product Manager Jim Gish; Senior Content Developer Alyssa Pratt; Senior Content Project Manager Jennifer Feltri-George; Developmental Editor Lisa Ruffolo, and Danielle Shaw, the MQA tester.

Writing a book is similar to entering a long-term relationship with an obsessive partner. Throughout the journey, life continues around you: teaching classes full time, presenting around the world, and celebrating family births and birthdays. As the world continues, those closest to you allow you to focus on your reclusive writing by assisting with every other task. My husband, Timothy, is credited with taking part in my nomadic life and most of all for his love, sous chef capabilities, and patience. Special thanks to my six children, Tim, Brittany, Ryan, Daniel, Breanne, and Eric, for providing much needed breaks filled with pride and laughter. To the newest members of my family, my grandsons Liam and Lochlan, who make me smile like no others, and yes we will go outside in just a minute. And a special thanks to Lisa Ruffolo as my developmental editor and master wordsmith who truly went above and beyond the call of duty with this fast-turnaround new project.

Voilà! Meet the Android

In this chapter, you learn to:

- ⦿ Understand the market for Android applications
- ⦿ Identify the role of the Android device in the mobile market
- ⦿ Describe the features of the Android phone
- ⦿ Identify which languages are used in Android development
- ⦿ Describe the role of Google Play in the mobile marketplace
- ⦿ Create an Android project using Android Studio
- ⦿ Explain the role of the Android project view
- ⦿ Specify the use of layout and widget controls in the user interface
- ⦿ Execute an Android application on an emulator
- ⦿ Open a saved Android project in Android Studio

Unless otherwise noted in the chapter, all screenshots are provided courtesy of Android Studio.

Welcome to the beginning of your journey in creating Android phone applications and developing for the mobile device market. Mobile computing has become the most popular way to make a personal connection to the Internet. Mobile phones and tablets comprise the fastest growing category of any technology in the world. Mobile phone usage has quickly outgrown the simple expectation of voice calls and text messaging. An average data plan for a mobile device, often called a smartphone, typically includes browsing the web, playing popular games such as Candy Crush or Words with Friends, using business applications, checking email, listening to music, recording live video, and mapping locations with GPS (global positioning system).

When purchasing a smartphone, you can choose from many mobile operating systems (OSs), including the iOS for the iPhone, Google Android, and Microsoft Phone. Recently, the Android phone has become the sales leader, outselling its competitors. The Android market is exploding with more than a million new activations of Android phones per day and over one billion total activations of Android devices now being used worldwide. Android users download more than 1.5 billion apps and games from Google Play each month. Nearly 73 percent of the world's mobile devices run on the Android platform. About 12.7 percent of the world's mobile devices run on the iOS platform.



IN THE TRENCHES

More than one-third of all U.S. households have canceled their landlines for the convenience of handling only one bill from a mobile carrier.

Creating mobile applications, called **apps**, for the Android phone market is an exciting new job opportunity. Whether you become a developer for a technology firm that creates professional apps for corporations or a hobbyist developer who creates games, niche programs, or savvy new applications, the Android marketplace provides a new means to earn income.



GTK

According to *U.S. News & World Report*, an App Developer is considered the top job in the technology field, earning a median salary of \$90,060.

Meet the Android

The Android phone platform is built on a free operating system primarily created by a company called Android, Inc. In 2005, Google obtained Android, Inc., to start a venture in the mobile phone market. Because Google intended the Android platform to be open source, the Android code was released under the Apache license, which permits anyone to download the full open-source Android code for free. Two years later, Google unveiled its first open-standards mobile

device called the Android (Figure 1-1). In less than a decade, the Android phone market has grown into the world's best-selling phone platform.

Android is the first open-source technology platform for mobile devices. Being an **open-source operating system** effectively means that no company or individual defines the features or direction of the development. Organizations and developers can extract, modify, and use the code for creating any app. The rapid success of the Android phone can be attributed to the collaboration of the **Open Handset Alliance** (openhandsalliance.com), an open-source business alliance of 80 firms that develop standards for mobile devices. The Open Handset Alliance is led by Google. Other members include companies such as Samsung, Sony, HTC, Texas Instruments, Kyocera, and LG. Google has produced their own phone, a modular device, under the name Project Ara. Competitors such as Apple, which produces the iPhone, do not have an open-source coding environment, but instead work with proprietary operating systems. The strength of the open-source community lies in the developers' ability to share their source code. Even though the open-source Android software is free, many developers are paid to build and improve the platform. For example, proprietary software such as the Apple operating system is limited to company employees licensed to build a program within the organization. The Android open-source platform allows more freedom so people can collaborate and improve the source code.

Many phone manufacturers install the Android operating system on their brand-name mobile phones due to its open-source environment. The open-source structure means that manufacturers do not pay license fees or royalties. With a small amount of customization, each manufacturer can place the Android OS on their latest devices. This minimal overhead allows manufacturers to sell their phones in the retail market for relatively low prices, often less than \$100. Low prices on Android mobile devices have increased the sales and popularity of these devices.

The open-source community also makes Android phones attractive for consumers. Android has a large community of developers writing apps that extend the functionality of the devices. Users, for example, can benefit from over 1.3 million apps available in the Android marketplace, many of which are free. Because the Android phone platform has become the leader in sales in the mobile market, the Android application market is expected to keep pace.



© iStock.com/deepblue4you

Figure 1-1 Android phone



GTK

On average, each smartphone worldwide runs 41 apps.

Android Phone Device

The Android phone is sold by a variety of companies under names you may recognize, such as Moto X, Galaxy, Droid, Xperia, OnePlus, Nexus, and HTC One (Figure 1-2).

4



© iStock.com/Maxiphoto



© iStock.com/deepbluelucy4you



© iStock.com/AX-image

Figure 1-2 Android models



IN THE TRENCHES

Android has ventured into the television market as well. Chromecast is a thumb drive that enables web streaming to television.



GTK

Chinese mobile phone manufacturers Huawei, Xiaomi, and Lenovo have a combined market share of four percentage points, according to technology research firm Gartner.

Android devices come in many shapes and sizes, as shown in Figure 1-2. The Android OS powers all types of mobile devices, including smartphones, tablets, netbooks, watches, MP4 players, automobiles, and Internet TVs. Android devices are available with a variety of screen dimensions. Many devices support a landscape mode where the width and height are spontaneously reversed depending on the orientation of the device. As you design Android apps, the screen size affects the layout of the user interface. To take full advantage of the capabilities of a particular device, you need to design user interfaces specifically for that device. For example, a smartphone and a tablet not only have a different physical screen size, but also different screen resolutions and pixel densities, which change the development process. As you develop an Android app, you can test the results on an **emulator**, which duplicates how the app looks and feels on a particular device. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. You can change the Android emulators to simulate the layout of a smartphone with a 4.5-inch screen or a tablet with a larger screen, both with high-density graphics. Android automatically scales content to the screen size of the device you choose, but if you use low-quality graphics in an app, the result is often a poorly pixelated display. As a developer, you need to continue to update your app as the market shifts to different platforms and screen resolutions.

The Android phone market has many more hardware case and size options than the 4-inch to 5.5-inch screen options of an iPhone. Several Android phones such as the Galaxy, Droid, and Nexus offer screens 4 inches or larger. This extra space is excellent for phone users who like to watch movies, play games, or view full web pages on their phone. In addition, many types of tablets, also called slates, run on the Android platform. Popular models of Android tablets include the Google Nexus, Samsung Galaxy, Amazon Kindle, and LG Pad. The Android tablets are in direct competition with other tablets and slate computers such as the iPad (various generations) and Windows tablets.



© iStock.com/Maxiphoto

Figure 1-3 Android tablet

Features of the Android

As a developer, you must understand a phone's capabilities. The Android platform offers a wide variety of features that apps can use. Some features vary by model. Most Android phones provide the features listed in Table 1-1.

Feature	Description
3-D graphics	The interface can support 3-D graphics for a 3-D interactive game experience or 3-D image rendering.
Facial recognition	Android provides this high-level feature for automatically identifying or verifying a person's face from a digital image or a video frame.
Front- and rear-facing camera	Android phones can use either a front- or rear-facing camera, allowing developers to create applications involving video calling.
Multiple language support	Android supports multiple human languages.
On-screen keyboard	The on-screen keyboard offers suggestions for spelling corrections as well as options for completing words you start typing. The on-screen keyboard also supports voice input.
Power management	Android identifies programs running in the background using memory and processor resources. You can close those apps to free up the phone's processor memory, extending the battery power. For optimized gaming, Android supports the use of a gyroscope, gravity and barometric sensors, linear acceleration, and rotation vector, which provide game developers highly sensitive and responsive controls.
Voice-based recognition	Android recognizes voice actions for calling, texting, and navigating with the phone.
Wi-Fi Internet tethering	Android supports tethering, which allows a phone to be used as a wireless or wired hot spot that other devices can use to connect to the Internet.

Table 1-1 Android platform features

Writing Android Apps

Android apps are written using the Java programming language. **Java** is a language and a platform originated by Sun Microsystems. Java is an **object-oriented programming language** patterned after the C++ language. Object-oriented programming encourages good software engineering practices such as code reuse. The official tool for Android application development, released in December 2014, is called **Android Studio**, an integrated development environment (IDE) for building and integrating application development tools and open-source projects. Android Studio IDE is exclusively dedicated to the purpose of creating Android applications. Prior to Android Studio, Eclipse ADT was the primary Android development environment. Eclipse developed applications in many programming languages, including Java, C, C++, COBOL, Ada, and Python.

As shown in the preface of this book, the first step in setting up your Android programming environment to install the free Android Studio IDE on your PC or Mac computer. The installation includes the Android **Software Development Kit (SDK)**, which runs in Android Studio. The Android SDK includes a set of development tools that help you design the interface of the program, write the code, and test the program using a built-in Android handset emulator. Another language called **XML** (Extensible Markup Language) is used to assist in the layout of the Android emulator.

Android Emulator

The Android emulator lets you design, develop, prototype, and test Android applications without using a physical device. When you run an Android program in Android Studio, the emulator starts so you can test the program. You can then use the mouse to simulate touching the screen of the device. The emulator mimics almost every feature of a real Android handset except for the ability to place a voice phone call. A running emulator can play video and audio, render gaming animation, and store information. Multiple emulators are available within the Android SDK to target various devices and versions from early Android phones onward. Developers should test their apps on several versions to confirm the stability of a particular platform. Android Studio includes a live layout editing mode that previews an app's user interface across a range of devices (Figure 1-4).



Figure 1-4 Android Studio live layout mode

The first Android version, release 1.0, was introduced in September 2008. Each subsequent version adds new features and fixes any known bugs in the platform. Android has adopted a naming system for each version based on sugary treats and dessert items, as shown in Table 1-2. After the first version, dessert names have been assigned in alphabetical order.

Version Name	Release Date
1.0 First version	September 2008
1.5 Cupcake	April 2009
1.6 Donut	September 2009
2.0 Éclair	October 2009
2.2 Froyo (Frozen Yogurt)	May 2010
2.3 Gingerbread	December 2010
3.0 Honeycomb	February 2011
4.0 Ice Cream Sandwich	May 2011
4.1 Jelly Bean	July 2012
4.4 Kit Kat	October 2013
5.0 Lollipop	November 2014

Table 1-2 Android version history



Critical Thinking

When creating an app, should I target the most current Android version only?

To support as many Android devices as possible, you should set the minimum development version to the lowest available that allows your app to provide its core feature set.

Getting Oriented with Market Deployment

The Android platform consists of the Android OS, the Android application development tools, and a marketplace for Android applications. After you write and test a program, you compile the app into an Android package file with the filename extension `.apk`. Programs written for the Android platform are sold and deployed through an online store called **Google Play** (*play.google.com*), which provides registration services and certifies that the program meets minimum standards of reliability, efficiency, and performance. Google Play requires that you sign an agreement and pay a one-time registration fee (currently \$25). After registration, you can publish your app on Google Play, provided the app meets the minimum standards. You can also release updates as needed for your app. If your app is free, Google Play publishes your app at no cost. If you want to charge for your app, the standard split is 70 percent of sales for the developer and 30 percent for the wireless carriers. For example, if you created an app for your city that featured all the top restaurants, hotels, attractions, and festivals and sold the app for \$1.99, you would net \$1.39 for each app sold. If you sell 5,000 copies of your app, you would earn almost \$7,000. You can use Google Play to sell a wide range of content, including downloadable content,

such as media files or photos, and virtual content such as game levels or potions (Figure 1-5). As an Android developer, you have the opportunity to develop apps for a fairly new market and easily distribute the app to the millions of Android mobile device owners. Google Play has nearly 300,000 more apps than the iTunes App Store—a difference of about 17 percent more.

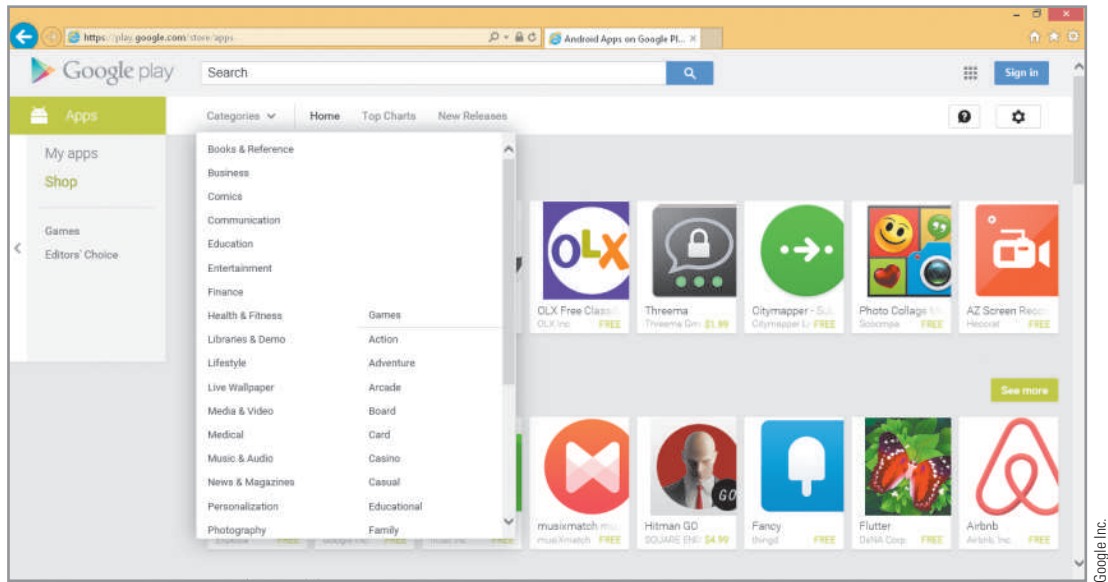


Figure 1-5 Google Play



IN THE TRENCHES

The Apple iTunes App Store charges a \$99 yearly registration fee to publish an app through the iPhone Dev Center. The iTunes App Store has a much more rigorous standards approval process than Google Play.

The online company Amazon also has a separate Appstore (amazon.com/appstore) where Android apps can be deployed and sold. The Amazon Appstore for Android is a category listed on Amazon.com. Customers can shop for apps from their PCs and mobile devices. The Amazon Appstore has an established marketing environment and search engine that displays a trusted storefront and creates app recommendations based on customers' past purchases. The Amazon Appstore charges a \$99 annual developer program fee, which covers application processing and account management for the Amazon Appstore Developer Program. Amazon also pays developers 70 percent of the sale price of the app; in addition, you can post free apps.

First Venture into the Android World

After installing Android Studio (developer.android.com/sdk) as instructed in the preface of this book, the next step is to create your first Android application. As programming tradition mandates, your first program is called Hello Android World. The following sections introduce you to the elements of Android Studio and provide a detailed description of each step to create your first app.