

Learn Android Studio 3 with Kotlin

Efficient Android App Development

Ted Hagos

Apress®

Learn Android Studio 3 with Kotlin: Efficient Android App Development

Ted Hagos
Manila, National Capital Region, Philippines

ISBN-13 (pbk): 978-1-4842-3906-3
<https://doi.org/10.1007/978-1-4842-3907-0>

ISBN-13 (electronic): 978-1-4842-3907-0

Library of Congress Control Number: 2018962941

Copyright © 2018 by Ted Hagos

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Steve Anglin

Development Editor: Matthew Moodie

Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484239063. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

For Adrienne and Stephanie.

Table of Contents

About the Author	xiii
About the Technical Reviewers	xv
Acknowledgments	xvii
Introduction	xix
Part I: The Kotlin Language	1
Chapter 1: Getting into Kotlin	3
About Kotlin.....	4
Installing the Java SDK	6
Installing on macOS.....	7
Installing on Windows 10.....	8
Installing on Linux	9
Installing Kotlin	10
Installing the Command Line Tools.....	10
Coding With the Command Line Tools	15
Installing IntelliJ	17
Creating a Project	19
The IntelliJ IDE	29
Chapter Summary	31
Chapter 2: Kotlin Basics	33
Program Elements	33
Literals.....	34
Variables.....	34
Expressions and Statements	36
Keywords.....	37

TABLE OF CONTENTS

- Whitespace 38
- Operators 39
- Blocks 41
- Comments 42
- Basic Types 44
 - Numbers and Literal Constants 44
 - Characters 46
 - Booleans 47
 - Arrays 47
 - Strings and String Templates 49
- Controlling Program Flow 51
 - Using ifs 51
 - The when Statement 53
 - The while Statement 55
 - for loops 55
- Exception Handling 57
- Handling Nulls 58
- Chapter Summary 60
- Chapter 3: Functions 63**
 - Declaring Functions 63
 - Single Expression Functions 67
 - Default Arguments 68
 - Named Parameters 69
 - Variable Number of Arguments 70
 - Extension Functions 71
 - Infix Functions 73
 - Operator Overloading 75
 - Chapter Summary 78

Chapter 4: Working with Types	79
Interfaces	79
Diamond Problem	81
Invoking Super Behavior.....	82
Classes.....	84
Constructors	85
Inheritance	88
Properties	92
Data Classes	96
Visibility Modifiers.....	100
Access Modifiers.....	102
Object Declarations.....	102
Chapter Summary	103
Chapter 5: Lambdas and Higher Order Functions	105
Higher Order Functions.....	105
Lambda and Anonymous Functions	109
Parameters in Lambda Expressions	110
Closures.....	113
with and apply	114
Chapter Summary	116
Chapter 6: Collections and Arrays	117
Arrays.....	117
Collections	121
Lists	123
Sets	124
Maps.....	125
Collections Traversal.....	127
Filter and Map.....	128
Chapter Summary	130

TABLE OF CONTENTS

- Chapter 7: Generics 133**
 - Why Generics 133
 - Terminologies..... 135
 - Using Generics in Functions..... 136
 - Using Generics in Classes..... 138
 - Variance 140
 - Subclass vs Subtype 144
 - Reified Generics 149
 - Chapter Summary 153

- Part II: Android Programming with Kotlin 155**

- Chapter 8: Android Studio Introduction and Setup..... 157**
 - History..... 157
 - Architecture 158
 - Android Studio IDE 160
 - Setup..... 161
 - Android Studio Configuration 163
 - Hardware Acceleration..... 169
 - Chapter Summary 170

- Chapter 9: Getting Started 173**
 - What’s in an App 173
 - Component Activation..... 176
 - Creating a Project 177
 - The IDE..... 190
 - Main Menu..... 192
 - Keyboard Shortcuts 193
 - Customizing Code Style..... 195
 - Chapter Summary 196

Chapter 10: Activities and Layouts	197
Application Entry Point.....	197
Activity Class	198
Layout File	200
View and ViewGroup Objects.....	201
Containers	203
Hello World	204
Modifying Hello World.....	208
Chapter Summary	218
Chapter 11: Event Handling	221
Introduction to Event Handling.....	221
Chapter Summary	237
Chapter 12: Intents	239
What Intents Are.....	239
Loose Coupling.....	242
Two Kinds of Intent	243
Intents Can Carry Data	243
Getting Back Results from Another Activity	246
Implicit Intents	249
Demo 1: Launch an Activity.....	251
Demo 2: Send Data to an Activity.....	259
Demo 3: Send and Get Data Back to and from an Activity	265
Demo 4: Implicit Intents	278
Chapter Summary	282
Chapter 13: Themes and Menus	283
Styles and Themes.....	283
Customizing the Theme	286
Menus	288
Chapter Summary	303

TABLE OF CONTENTS

- Chapter 14: Fragments 305**
 - Introduction to Fragments..... 305
 - Book Title and Description, a Fragments Demo 311
 - Fragments Demo, Dynamic 337
 - Chapter Summary 341

- Chapter 15: Running in the Background..... 343**
 - Basic Concepts 344
 - The UI Thread 344
 - Threads and Runnables 349
 - Using the Handler Class 354
 - AsyncTask 357
 - Anko’s doAsync..... 360
 - A Real-World Example 363
 - Chapter Summary 371

- Chapter 16: Debugging 373**
 - Syntax Errors 373
 - Runtime Errors 377
 - Logic Errors..... 382
 - Walking Through Code..... 385
 - Other Notes 387
 - Chapter Summary 388

- Chapter 17: SharedPreferences..... 389**
 - Sharing Data Between Activities..... 398
 - Chapter Summary 406

- Chapter 18: Internal Storage 407**
 - Overview of File Storage 407
 - Internal and External Storage..... 408
 - Cache Directory 409

How to Work with Internal Storage	409
Chapter Summary	424
Chapter 19: BroadcastReceivers	425
Introduction to BroadcastReceivers	425
System Broadcast vs. Custom Broadcast	426
Manifest Registration vs. Context Registration	427
Basics of BroadcastReceivers.....	430
Implicit vs. Explicit Broadcast Actions	432
Demo App: Custom Broadcast.....	433
Demo App: System Broadcast.....	440
Other Notes	443
Chapter Summary	444
Chapter 20: App Distribution	445
Preparing the App for Release	446
Prepare Materials and Assets for Release.....	446
Configure the App for Release.....	447
Build a Release-Ready Application.....	448
Releasing the App	452
Chapter Summary	456
Index.....	459

About the Author

Ted Hagos is the CTO and Data Protection Officer of RenditionDigital International, a software development company based out of Dublin, Ireland. Before he joined RDI, he had various software development roles and also spent time as trainer at IBM Advanced Career Education, Ateneo ITI, and Asia Pacific College. He spent many years in software development dating back to Turbo C, Clipper, dBase IV, and Visual Basic. Eventually, he found Java and spent many years there. Nowadays, he's busy with full-stack JavaScript and Android.

About the Technical Reviewers



Massimo Nardone has more than 24 years of experience in Security, Web/Mobile development, Cloud, and IT Architecture. His true IT passions are Security and Android.

He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years.

He holds a Master of Science degree in Computing Science from the University of Salerno, Italy.

He has worked as a Project Manager, Software Engineer, Research Engineer, Chief Security Architect, Information

Security Manager, PCI/SCADA Auditor and Senior Lead IT Security/Cloud/SCADA Architect for many years.

Technical skills include: Security, Android, Cloud, Java, MySQL, Drupal, Cobol, Perl, Web and Mobile development, MongoDB, D3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, Django CMS, Jekyll, Scratch, etc.

He worked as visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and Proxy areas).

He currently works as Chief Information Security Officer (CISO) for Cargotec Oyj, and he is a member of ISACA Finland Chapter Board.

Massimo has reviewed more than 45 IT books for different publishers and has coauthored *Pro JPA in Java EE 8* (Apress, 2018), *Beginning EJB in Java EE 8* (Apress, 2018), and *Pro Android Games* (Apress, 2015).

ABOUT THE TECHNICAL REVIEWERS



Val Okafor is a software architect with expertise in Android development and resides in sunny San Diego, California. He has over 12 years of industry experience and has worked for corporations such as Sony Electronics, The Home Depot, San Diego County, and American Council on Exercise. Val earned his BSc in IT from National University, San Diego and his Masters in Software Engineering from Regis University, Colorado. He is the creator and principal engineer of Pronto line of mobile apps including Pronto Diary, Pronto Invoice, and Pronto Quotes.

His passion for software development goes beyond his skill and training; he also enjoys sharing his knowledge with other developers. He has taught Android development to over 5,000 students through Udemy, and his blog valokafor.com is considered an essential reading for Android developers. Val was also recently named among the first cohort of Realm MVP program because of his active participation in the Realm database community.

Acknowledgments

To Stephanie and Adrienne, for bearing with me for the past 9 months while I wrote this book. My thanks and my love.

To Mark Powers, for his understanding when I missed some of the writing deadlines and for keeping the schedule straight.

To Steve Anglin, for bringing me to Apress.

To everyone who made this book possible, Thank you. It truly feels great to hold one's printed book in one's hands. It's even more awesome the second time around.

Introduction

Welcome to the Kotlin edition of *Learn Android Studio 3*. This book will help you get started in your programming journey with the little green robot. You already bought the book, so you don't need to be convinced that programming for the mobile platform offers a lot of opportunity for software developers. Thank you for buying it, by the way.

Who This Book Is For

The book is aimed at beginning Android programmers, but it isn't for people who are completely new to programming. Ideally, you already are a Java programmer trying to get your feet wet in Android, and you wanna try the Kotlin language (coz all your dev friends told you it was cool). But in case you're not a Java developer or you don't have Android programming experience, don't sweat it. The book is friendly enough—I tried hard to write it that way—and approachable enough such that anyone with a passing knowledge of either C#, JavaScript, C, or C++ will be able to follow the code samples and the concepts presented in this book.

What's Different in the Kotlin Edition

All the code examples and the demo projects are mostly new. They're not a plain Kotlin port of the first edition's examples. I've also added new chapters; here they are:

- Collections
- Generics
- Higher Order Functions
- Broadcast Receivers

Some chapters in the first edition have been split into two or more chapters. I split them so that I can treat the subjects with more depth—for example, “Intents,” “SharedPreferences,” “Internal Storage,” and “Fragments.”

Organization and Treatment

The book is divided into two major parts. Chapters 1 to 7 are all about the Kotlin language, and Chapters 8 to 20 are about Android programming.

While you can use it as a reference book, I didn't write it that way. It's not meant as a substitute for the docs in <https://kotlinlang.org> or the Android developer guides <https://developer.android.com>. It's also not meant to be a "Definitive Guide" type of book where you can spend hours or days exploring every nook and cranny. Quite the contrary—I wanted it to be a "get started quick" type of book, like a recipe book, but without losing our grasp on the fundamental concepts.

Android and Kotlin are big subjects; I don't think there exists a "single best way" to present the materials for either of these two. So, I made certain bets on the instructional design. Here they are:

- **Bite-sized concepts.** The troublesome topics are broken down into a series of small steps so that you can solve them in isolation. When you can solve small problems, it gives you confidence to solve bigger ones. This approach helps a beginning programmer to grow in the direction of skill.
- **Conciseness.** I tried to keep each chapter as short as possible, so you can finish it in one sitting. Originally, I wanted each chapter to be a "20-minute read"; that was too ambitious, so, I gave up on it—but still, the chapters are short.
- **Multiple Learning Curves.** The book is about three topics: Android Studio, Android Programming, and Kotlin. Although Kotlin and Android programming may seem to have dedicated chapters for them, techniques on how to use Android Studio (and IntelliJ) are scattered throughout the book.
- **Balance between concept and code.** Admittedly, the treatment is biased (just a little bit) toward code. Programming is not a spectator sport; we learn by doing. Nonetheless, in every chapter, I tried to explain what the fundamental concepts are, what we're trying to do, what problems are we trying to solve, how we might solve those problems, and what does the solution look like—in code. Almost all of the chapters have one or more demo projects in them.

- **Verbose and complete code presentations.** Sometimes (most of the time actually), I presented the full source example, but only one or two lines of it are relevant. I erred on the side of caution (and verbosity) because it's easier for a beginner to understand the relevant codes if he can see it in relation to the whole program. So, you don't have to worry about, "Where do I put this code? Does this go inside function main or inside a class?"
- **Immediacy and coherence.** Like I said, I wanted this to be a "get started quick" or a "recipe" kind of book. So, instead of covering everything, including the kitchen sink, I chose to cover some topics and ignore others. I chose use-cases whose complexities are easy or moderate and covered topics that are only relevant for those use-cases. For example, in the `BroadcastReceiver` and `Intent` chapters, I didn't cover `LocalBroadcastManager` and `PendingIntent`. Cool as these topics are, they weren't relevant for the use-cases I chose. If I added more use-cases or demo-projects, that would have stretched the length of the chapter. It's a balancing act, you see.
- **Independent demo projects.** I designed them as such so that the demo project could be started (and followed) from scratch. There is no "putting it all together" project in the end. This way, the book can be conveniently used as a reference. If you pick a topic, it's almost self-contained, including the demo project.

In the end, I can only hope that the bets I made will pay off and that you will walk away as a slightly better programmer after reading the book.

Chapter Overviews

Chapter 1: "Getting into Kotlin" introduces the language. It tells you how to setup Kotlin in various ways on the three major platforms: macOS, Linux, and Windows. It also contains instructions on how to create, configure, and run a project in IntelliJ—this is the IDE I used to create all the Kotlin code samples for Chapters 1 through 7.

Chapter 2: "Kotlin Basics" dives into the language fundamentals of Kotlin. You'll learn the basic building blocks of a Kotlin program (e.g., Strings, control structures,

exception handling, basic data types). You'll also see some of Kotlin's features that are very different from Java, like its treatment of nullable and non-nullable types.

Chapter 3: "Functions." There's a whole chapter dedicated to functions because Kotlin's functions have something new up their sleeves. It has all the trimmings of a modern language like default and named parameters, infix functions, and operators; and with Kotlin, we can also create extension functions. Extension functions lets you add behavior to an existing class, without inheriting from it and without changing its source.

Chapter 4: "Working with Types." This chapter deals with object-oriented topics. You'll learn how Kotlin treats interfaces, classes, and access modifiers. We'll also learn about the new *data classes* in Kotlin. It also talks about *object declarations*—it's the replacement for Java's *static* keyword.

Chapter 5: "Lambdas and Higher Order Functions." Now we go to Kotlin's functional programming capabilities. It discusses how to create and use higher-order functions, lambdas, and closures.

Chapter 6: "Collections" walks through the classic collection classes of Java and how to use them in Kotlin.

Chapter 7: "Generics." Using generics in Kotlin isn't that much different from Java. If generics is old hat for you, then most of this chapter will be a review. But try to read through it still because it talks about *reified generics*, which Java doesn't have.

Chapter 8: "Android Studio Introduction and Setup." This chapter talks a bit about Android's history, its technical make-up, and the OS. It also walks you through the installation and setup of Android Studio.

Chapter 9: "Getting Started" gets you grounded on the fundamental concepts about Android programming. It talks about components, what they are, how they are organized, and how they come together in an Android app. In this chapter, you'll learn how the basic workflow of an Android project—how to create a project and run it on an emulator

Chapter 10: "Activities and Layouts." Here, we'll learn how to build a UI. Activity, Layout, and View objects are the building blocks for an Android UI.

Chapter 11: "Event Handling." You'll learn how to react to user-generated events like clicks and longclicks. We'll use some concepts that we learned in Chapters 4 and 5 (inner objects and lambdas) to help us write more compact and succinct event-handling code.

Chapter 12: "Intents." This chapter reviews some fundamental concepts on Android programming, specifically the concept of components, which dovetails to the topic of Intents. You'll learn how to use Intents to launch another Activity and pass data in-and-around Activities.

Chapter 13: “Themes and Menus.” This is a short chapter. You’ll learn how to add styles/themes to your app. We’ll also work with some menus and the ActionBar.

Chapter 14: “Fragments.” You’ll learn how to use Android Fragments as a more granular composition unit for UI. We’ll also see how to use Fragments to address changes in device orientation.

Chapter 15: “Running in the Background.” Any non-trivial app will do something substantial like read from a file, write to a file, download something from the network, etc. These activities will likely take more than 16 ms to execute (you’ll learn why 16 ms should be the upper limit and why you should not exceed it). When that happens, the user will see and feel “jank.” This chapter discusses the various ways on how to run our code in a background thread.

Chapter 16: “Debugging” shows some of the things you can do to debug your apps in Android Studio 3. It goes through a list of the kinds of errors you might encounter while coding and what you can do in Android Studio to respond them.

Chapter 17: “SharedPreferences.” When you need to save simple data, you can use the SharedPreferences API. This chapter walks you through detailed examples on how to do that.

Chapter 18: “Internal Storage.” Just like in SharedPreferences, you can also store data using the Internal Storage API of Android. This chapter discusses internal and external storage.

Chapter 19: “BroadcastReceivers.” Android has a way to make highly decoupled components talk to each other. This chapter talks about how BroadcastReceivers can facilitate messaging for Android components.

Chapter 20: “App Distribution.” When you’re ready to distribute your app, you’ll need to sign it and list it in a marketplace like Google Play. This chapter walks you through the steps on how to do it.

How to Get the Most From This Book

I designed it like a workbook; it’s best to use it like that. Most chapters have a “Demo Project” section. There are details on how to create a project—for example, what name should you use for the project, the minimum SDK to target, etc. The reason I included this information is so you can follow the coding exercise.

INTRODUCTION

I used three kinds of blocks in the book: *Examples*, *Listings*, and *Figures*.

- **Examples** are commands that you would type in a terminal window.
- **Listings** contains program or code listing; it's something that you would type in a program file.
- **Figures** could be screenshots or diagrams. Some of the screenshots are annotated to point out a sequence of steps and how to do them on the IDE. I used Android Studio 3.1 and IntelliJ 2018.2 for the examples in this book; it's possible that by the time you read this book, you'll be using a different or higher version of these tools.

Programmers (mostly) learn by doing. If you work your way through the demo projects, I think the lessons will stick better. Remember that coding is like swimming or driving, you can read as many books as you want on the subjects, but if you don't go in the water or behind the wheel, you won't get anywhere.

Source Code

Source Code for this book can be accessed by clicking the **Download Source Code** button at www.apress.com/9781484239063.