

My First Mobile App for Students

*A comprehensive guide to Android app
development for beginners*

Zaid Kamil



www.bpbonline.com

Copyright © 2024 BPB Online

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor BPB Online or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

BPB Online has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, BPB Online cannot guarantee the accuracy of this information.

First published: 2024

Published by BPB Online

WeWork

119 Marylebone Road

London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55515-933

www.bpbonline.com

Dedicated to

My Mother

Farzana Hafeez

About the Author

Zaid Kamil is a Google-certified Android app developer and a professional coding trainer who has been developing and teaching how to make Android apps for over a decade. He is the Training Head at Digipodium, a leading IT training institute in Uttar Pradesh, where he mentors and guides hundreds of students and professionals in various domains of technology. He has also conducted many workshops and seminars at various universities and colleges. He has a rich portfolio of projects in both Java and Kotlin, using Android and various third-party libraries and APIs. He is also skilled in area of Data Science, AI, Cloud services and Web development, and holds multiple certifications from IBM and Microsoft in Python. He has a keen interest in designing and implementing machine learning models to analyze data and improve business operations. He has worked with cross-functional teams to design and deploy AI solutions for various domains. Zaid Kamil is passionate about learning new skills and technology. He constantly updates himself with the latest trends and developments in the tech industry. He also likes to explore new tools and frameworks that can enhance his coding abilities. He believes that learning is a lifelong process and that one should never stop growing and improving oneself.

About the Reviewer

In the ever-evolving landscape of technology, **Mukund Soni** emerges as a dynamic force, bridging the realms of academia, creativity, and practical application. A testament to his dedication, Mukund has achieved remarkable grades, reflecting his unwavering commitment to academic brilliance. His thirst for innovation finds expression in his active involvement in hackathons and technological competitions, where he consistently thrives.

Currently, Mukund crafts his unique narrative in the digital sphere. His forte lies in Frontend development, seamlessly blending the artistic with the utilitarian through platforms like Next.js and TypeScript. His proficiency in Web3 and Blockchain integration unveils new vistas of possibilities, as he engineers innovative solutions. His artistic prowess manifests in AI image generation, while his practicality shines through in Content Management Systems development.

His earlier tenure included delving into intricate projects. This experience not only nurtured his skills but also exposed him to the intricacies of Microsoft Azure, Hybrid Device Management, Network Protection, and Vulnerability Detection.

Mukund's zeal for growth extends to personal projects, a testament to his drive to impact the world positively. He is currently developing a Blind Assistance Application, which endeavours to empower the visually impaired by leveraging technology to enhance their daily lives. In the heart of his capabilities lies a profound understanding of various technologies, ranging from Python and Next.js to Machine Learning, Microsoft Azure, and Blockchain. Mukund Soni's journey is a symphony of technical prowess, innovative thinking, and an unwavering commitment to excellence.

This visionary individual is not just a product of technology; he is a creator, an innovator, and a catalyst of change. In a world marked by digital transformation, Mukund Soni is a beacon of potential, a harbinger of what is yet to come.

Acknowledgements

As I pen down these words, my heart swells with gratitude for the incredible individuals who have been a part of this journey. To my unwavering family and friends, you have been my steadfast pillars of support. My parents, in particular, have been the guiding lights, illuminating the path to my dreams. In the midst of ideation and creation, my best friend Bharvi emerged as a wellspring of creativity. Your ideas breathed life into the apps, and for that, I am truly thankful.

The path from concept to completion would not have been possible without the expertise of BPB Publications. Their mentorship, expert guidance and unwavering belief, infused it with a sense of purpose, transforming this vision into a tangible reality. Their expertise has been truly instrumental.

I am also humbled by the contributions of my colleagues and co-workers and my dear students. Each interaction, each collaboration, has been a learning experience that is indelibly etched into my journey.

Lastly, my heartfelt gratitude goes to you, dear readers. Your interest in my book, your engagement with its content, your encouragement – these are the threads that have woven this narrative into a reality. This book is as much yours as it is mine. Thank you for being a part of this odyssey of creation and discovery.

Preface

In today's tech-driven world, mobile apps are how businesses and people connect with each other. Android OS, with a massive 71% share of the global market and a whopping 2.87 billion apps, is where you want to start if you dream of creating your own apps. This book is your guide, showing you step by step how to go from an idea to making real apps.

We start at the beginning, covering the basics you need to know. We will guide you through setting up the tools you will use, like Android Studio, and teach you about Kotlin, the language used for creating Android apps. By the end, you will have a solid foundation to build upon.

Learning is best when you actually do things, so this book is all about hands-on experience. You will create six different apps, each with its own unique design and coding. This way, you will turn what might seem complex into something practical and understandable. As you work through the chapters, you will pick up new skills. You will learn why Kotlin is better than Java for app making, and you will become comfortable with Android Studio. You will also get the hang of designing the look of your apps and making them work smoothly.

By the end of this journey, you will be equipped with the knowledge to build apps with confidence. You will understand the ins and outs of app creation, from the basics like setting up the environment to more advanced topics like connecting to the cloud. Whether you are just starting out or already have some tech experience, this book is your companion in your adventure of creating Android apps.

Chapter 1: Welcome, Future App Developer

Begin your journey with the Android platform's evolution and essential tools. Explore Android Studio, your primary software for development. Learn to download, install, and troubleshoot common issues. Learn how to work with the Kotlin programming language with the help of basics given in the chapter.

Chapter 2: App 1—Dynamic Dice Simulator (Part 1)

Divided into four sections, this chapter guides you through creating a new Android project, understanding Android Studio's interface and project structure, comprehending app architecture, using the design and XML editors, and implementing buttons, text, and images.

Chapter 3: App 1—Dynamic Dice Simulator (Part 2)

This chapter extends the previous app development, introducing wireframes of the completed app, multiple activities, the Android activity lifecycle, and setting up splash screens. Learn to handle intents, data passing, and add animation using MotionLayout.

Chapter 4: App 2—State Trivia

Craft a new app project focusing on state trivia. Explore collections, data classes, resources, themes, and applying the Material 3 theme. Dive into RecyclerView, data binding, and the Android menu system. Implement implicit intents for external app launches.

Chapter 5: App 3—Movie Booking

Discover Android Jetpack’s Navigation Architecture and Jetpack components for creating robust, efficient apps. Understand fragments, navigation components, ViewModel, LiveData, and DataBinding. Create a new project with Basic Activity, design app fragments, manage flow, and handle data passing.

Chapter 6: App 4—Book Finder

Uncover background tasks using coroutines in Kotlin. Utilize Retrofit and Coil libraries to collect data from Google Books API. Master multithreading, set up the project, handle permissions, and parse data using Moshi and RecyclerView. Learn to display internet-received images with Coil.

Chapter 7: App 5—Flash Cards (Part 1)

Learn data persistence through databases. Build a flashcards app with ViewPager2. Set up fragments, Navigation graph, ViewModel, and LiveData. Understand SQL concepts, queries, and set up the Room library. Dive into Entity, DAO, ViewModel, and ViewModelFactory.

Chapter 8: App 5—Flash Cards (Part 2)

Continue the Flash Cards app, integrating different parts. Create a form to save and modify database entries. Utilize ViewPager2 to display cards from the database. Explore ListAdapter for memory optimization and grasp extension functions in Kotlin.

Chapter 9: App 6—Inspire Me

Conclude your journey with cloud database and authentication concepts. Explore Firebase, linking Android Studio, and Firebase operations. Establish an authentication system, upload text data to the cloud, and receive live updates.

Bonus: Learn to upload your projects to GitHub.

Code Bundle and Coloured Images

Please follow the link to download the *Code Bundle* and the *Coloured Images* of the book:

<https://rebrand.ly/ns791wq>

The code bundle for the book is also hosted on GitHub at **<https://github.com/bpbpublications/My-First-Mobile-App-for-Students>**. In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **<https://github.com/bpbpublications>**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Did you know that BPB offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.bpbonline.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at :

business@bpbonline.com for more details.

At **www.bpbonline.com**, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on BPB books and eBooks.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at business@bpbonline.com with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit www.bpbonline.com. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit www.bpbonline.com.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

<https://discord.bpbonline.com>



Table of Contents

1. Welcome, Future App Developer.....	1
Introduction.....	1
Structure.....	1
Objectives.....	2
History of Android OS.....	2
History of Kotlin.....	4
Market impact of Mobile OSes	5
Setting up Android Studio	6
<i>Windows installation</i>	8
<i>Mac installation</i>	8
<i>Linux installation</i>	8
Android SDK setup	9
Kotlin classes and objects	12
<i>Class in Kotlin</i>	14
<i>Object in Kotlin</i>	16
Functions and variables in Kotlin	18
<i>Function</i>	18
<i>Variables</i>	22
Conditions and loops	24
<i>If expression</i>	24
<i>When expression</i>	25
<i>Loops</i>	26
<i>For</i>	26
<i>While</i>	27
Conclusion.....	28
Points to remember	28
Exercises.....	29
2. App 1—Dynamic Dice Simulator (Part 1)	31
Introduction.....	31
Structure.....	31
Objectives.....	32

Android project wizard	33
Android studio interface	35
Android app project structure	38
Android design editor interface	41
Adding drawables	42
View	43
Viewgroups.....	44
ConstraintLayout.....	44
Adding TextView.....	46
Adding button.....	48
Adding ImageView	49
XML editor.....	52
Editing attributes of views	52
Coding Kotlin function	53
Activity and types of activities	55
Event listener.....	57
Emulator.....	59
Developer mode in Android phones	61
Running the app	62
Gradle build.....	63
Conclusion.....	64
Exercises	64
3. App 1—Dynamic Dice Simulator (Part 2).....	65
Introduction.....	65
Structure.....	65
Objectives.....	66
Multiple activities in app.....	66
Intent.....	69
Types of Intent.....	69
Companion object in Kotlin	71
Passing data through intent	71
Intent filter	72
Android manifest.....	72
Motion layout.....	74

Motion editor.....	82
Motion scene.....	83
Transition	83
Finishing up the app	85
Conclusion.....	87
Exercises.....	88
4. App 2—State Trivia.....	89
Introduction.....	89
Structure.....	89
Objectives.....	90
Collection and generics.....	90
Model and data class.....	92
<i>Data class</i>	92
<i>Create the new app</i>	93
<i>Data source</i>	93
Material design	95
<i>Updating to Material 3 theme</i>	96
Changing the app icon.....	99
ViewBinding.....	100
<i>Enabling the ViewBinding</i>	101
Gradle sync.....	102
Creating the scrollable list.....	102
RecyclerView	103
<i>Single element layout</i>	105
CardView	106
Adapter for RecyclerView	108
<i>Abstract methods</i>	109
<i>Constructor for the adapter</i>	110
<i>ViewBinding inside the adapter</i>	110
<i>Passing the data to ViewHolder</i>	110
<i>Setting the size of items</i>	111
ViewHolder	111
<i>The full Adapter class code</i>	112
<i>Completing the RecyclerView</i>	113

<i>Changing the layout</i>	114
Options menu.....	115
The click functionality.....	117
Conclusion.....	118
Exercises.....	119
5. App 3—Movie Booking	121
Introduction.....	121
Structure.....	121
Objectives.....	122
What is fragment?.....	122
Fragment lifecycle.....	124
The basic activity	125
NavHostFragment.....	127
Android jetpack	128
Creating a new project	129
Setting up the fragments.....	130
<i>Rename existing fragment</i>	131
<i>Create a new package</i>	131
<i>Creating new fragments</i>	133
<i>Update fragment labels</i>	135
Kotlin getter and backing properties	136
Updating fragment default code.....	138
<i>Fix new fragments</i>	139
<i>Create a user interface for each fragment</i>	141
<i>fragment_intro.xml</i>	141
<i>fragment_day_select.xml</i>	142
<i>fragment_time_select.xml</i>	143
<i>fragment_seat_select.xml</i>	145
<i>fragment_payment.xml</i>	146
<i>fragment_confirm.xml</i>	147
<i>fragment_receipt.xml</i>	148
Navigation graph.....	148
MVVM architecture in Android	151
DataBinding.....	153

<i>Updating the build.gradle file</i>	153
ViewModel.....	154
<i>Shared ViewModel</i>	155
<i>Creating the ViewModel</i>	156
LiveData.....	156
<i>Immutable versus mutable LiveData</i>	157
<i>Updating the ViewModel</i>	157
<i>ViewModel + DataBinding in Layout files</i>	162
<i>fragment_intro.xml</i>	163
<i>fragment_day_select.xml</i>	164
<i>fragment_time_select.xml</i>	165
<i>fragment_seat_select.xml</i>	166
<i>fragment_payment.xml</i>	167
<i>fragment_confirm.xml</i>	168
<i>fragment_receipt.xml</i>	168
Lifecycle observer.....	168
<i>Completing the fragments</i>	169
<i>IntroFragment.kt</i>	169
<i>DaySelectFragment.kt</i>	170
<i>TimeSelectFragment.kt</i>	171
<i>SeatSelectFragment.kt</i>	172
<i>PaymentFragment.kt</i>	173
<i>ConfirmFragment.kt</i>	174
<i>ReceiptFragment.kt</i>	175
Running the app	175
Conclusion.....	177
Exercises.....	177
6. App 4—Book Finder.....	179
Introduction.....	179
Structure.....	179
Objectives.....	180
Multithreading.....	180
Coroutines.....	182
Web services and networking.....	183

The Google Books API	184
Setting up the new project.....	188
Retrofit service.....	189
Moshi and data conversion.....	191
Permission in Android.....	193
Creating the Project UI.....	193
<i>Update strings.xml</i>	193
<i>Update navigation graph</i>	193
<i>Update activity_main.xml layout</i>	193
<i>Change fragment_search.xml layout</i>	194
<i>Change fragment_result.xml</i>	195
<i>Edit the book_card_view.xml</i>	196
Integrating the code.....	197
<i>Creating the BookViewModel</i>	197
<i>Editing the SearchFragment</i>	200
<i>Editing the ResultFragment</i>	202
<i>Handling the BookAdapter</i>	203
BindingAdapter	206
Adding the data binding to layout.....	207
Running the app	209
Conclusion.....	209
Exercises	210
7. App 5—Flash Cards (Part 1)	211
Introduction.....	211
Structure.....	211
Objectives.....	212
The concept of database.....	212
<i>Local databases</i>	212
<i>Remote databases</i>	213
Table, columns and rows	213
<i>Tables</i>	214
<i>Columns</i>	214
<i>Rows</i>	215
Datatypes	215

Primary key	216
SQL and basic queries	217
<i>SELECT</i>	218
<i>INSERT</i>	218
<i>UPDATE</i>	218
<i>DELETE</i>	219
Room library	219
Creating the new project	221
Adding room dependency	224
Entity in room	225
Data Access Object	226
<i>Creating the database</i>	228
<i>Attaching the database with application</i>	229
Conclusion	231
Exercises	231
8. App 5—Flash Cards (Part 2)	233
Introduction	233
Structure	233
Objectives	234
Fetching data using flow	234
ViewModel and DAO	235
ViewModelFactory and DAO	239
Adding data	240
<i>The layout design</i>	240
<i>Add the resource strings</i>	242
<i>The fragment's code</i>	242
Setting the home screen with ViewPager	246
<i>ViewPager2</i>	246
<i>The layout design</i>	246
<i>The fragment's code</i>	248
Setting up the ListAdapter	252
Adding more functionality	255
View and delete item	256
<i>The layout</i>	256

<i>Kotlin code</i>	259
<i>Navigation graph implementation</i>	261
Editing item	261
<i>The layout</i>	261
<i>Kotlin code</i>	263
<i>Navigation graph implementation</i>	265
Running the app	266
Conclusion.....	267
Exercise.....	267
9. App 6—Inspire Me	269
Introduction.....	269
Structure.....	269
Objectives.....	270
Cloud services	271
Setting up the project	272
<i>Fragments</i>	273
<i>ViewModels</i>	273
<i>Adapter</i>	273
<i>Navigation graph</i>	274
<i>Layouts</i>	274
<i>Model</i>	274
<i>Drawables</i>	275
<i>BindingUtil.kt</i>	275
<i>Menu</i>	276
<i>Strings.xml</i>	276
App’s project structure.....	277
Firebase integration.....	278
Firebase console	281
Firebase authentication.....	282
Firebase Firestore.....	282
Project UI and layout.....	283
<i>activity_auth.xml</i>	283
<i>fragment_login.xml</i>	284
<i>fragment_register.xml</i>	290

<i>post_card.xml</i>	295
<i>fragment_display.xml</i>	298
<i>fragment_add.xml</i>	301
ViewModels.....	305
<i>CloudAuthViewModel</i>	305
<i>FirebaseViewModel</i>	309
Navigation graph.....	313
Fixing the menu	314
Adapter	315
Implementation of fragments	317
<i>LoginFragment</i>	317
<i>RegisterFragment</i>	320
<i>DisplayFragment</i>	323
<i>AddFragment</i>	327
Uploading apps to GitHub.....	330
Running the app	331
Conclusion.....	332
Exercises.....	332
Index	333-339

CHAPTER 1

Welcome, Future App Developer

Introduction

Welcome, developer of the future! You are reading this chapter so that you can get an idea about the history of the Android operating system and the different versions of the Android operation system, and its impact on the mobile OS market. Then you are going to understand how to correctly install Android studio and its SDK on your pc/laptop. Finally, you will learn the language Kotlin, the official language for developing Android apps. To summarize, this chapter will provide you with a basic idea about Android, installation of Android studio, and coding in Kotlin. Concentrate on the coding portion of Kotlin, which will be very useful, as we are going to develop a lot of Android apps through each chapter. So, let us get started.

Structure

In this chapter, we will cover the following topics:

- History of Android OS
- History of Kotlin
- Market impact of Mobile OSes
- Setting up Android Studio

- Android SDK setup
- Kotlin classes and objects
- Functions and variables
- Condition and loops

Objectives

By the end of this chapter, the reader will know about the Android OS history, the market impact of Android, how to setup Android Studio IDE, and setup the software for developing Android apps. Finally, the reader will learn about Kotlin language and its main concepts, such as classes and objects, functions, conditions, and loops.

History of Android OS

It has been more than a decade since the first Android phone came out, and with it came the Android operating system. However, the real beginning of Android was in the year 2003, well before Apple iOS. Android Inc was founded by Andy Rubin, Rich Miner, Nick Sears and Chris White, and their aim was to develop *smarter mobile devices that are more aware of its owner's location and preferences*. Originally, the Android OS was meant to improve digital cameras but due to declining digital camera market, Android Inc, changed their target to mobile phones. The biggest change, however, came in 2005, when Google bought the Android Inc. company.

All the founding members continued developing the OS under the ownership of Google. Google made Android an open-source operating system that allowed it to become highly popular with third-party phone manufacturers. In September 2008, the first Android smartphone was released: T-Mobile G1, or the HTC Dream, with Android 1.0, and the rest became history. Over the years, various Android versions were released, and every major Android OS update had a special public name based on deserts. Following is the list of major Android versions and what they bring to the table:

Version	Release year	Features
Android 1.0–1.1	2008	Basic app support, support for Wi-Fi, Bluetooth, Web browsing, and camera.
Android 1.5 Cupcake	2009	Introduced auto rotation in phones, support for third-party keyboards, and the ability to upload YouTube videos.

Version	Release year	Features
Android 1.6 Donut	2009	Added CDMA networks, quick google search box, camera/video toggle system, and introduced power widgets for Wi-Fi, Bluetooth, and GPS.
Android 2.0–2.1 éclair	Oct, 2009	Google released version 2.0 of Android OS and named it Eclair. It added text to speech, live wallpapers, multiple accounts, and Google map navigation system.
Android 2.2 Froyo	May, 2010	Wi-Fi hotspot functionality, push notification, and flash support.
Android 2.3 Gingerbread	Sep, 2010	UI got a major overhaul, added support for Near Field Communication (NFC) to allow sharing, and also added support for multiple cameras.
Android 3.0 Honeycomb	Feb, 2011	Support for Android Tablet screens and included the new concept of fragmented sections in app, that could be arranged responsively for small and large screens, and both.
Android 4.0 Ice Cream Sandwich:	Oct, 2011	Added support for face-unlock (biometric sign-in), on-screen buttons for navigation, swipe gesture, and data usage monitor.
Android 4.1–4.3 Jelly Bean	2012	Notification action buttons, full Google Chrome browser support, introduction of Google now, and screen casting support.
Android 4.4 KitKat	2013	Now run on a minimum of 512 MB of RAM, which allowed manufacturers to make cheaper Android smartphones, and it led to increased market share.
Android 5.0 Lollipop	2014	New UI design component called Material Design. This new UI was very well received by the users.
Android 6.0 Marshmallow	2015	Support for USB-C, the introduction of Google Pay, fingerprint recognition support, and a vertically scrolling app drawer.
Android 7.0 Nougat	2016	Better multitasking system, split-screen mode, better 3D graphic support, and Vulkan API for graphics.
Android 8.0 Oreo	2017	New settings menu, picture in picture mode, notification grouping, and autofill system for forms.
Android 9.0 Pie	2018	New gesture-controlled slim button at the bottom center. Battery performance improvement using on-device Machine learning and features such as shush and app slices.

Version	Release year	Features
Android 10	2019	Google ditches the naming convention, new development API, dark mode, gesture navigation control, new sharing menu, smart reply, and app-based permissions.
Android 11	2020	record screen, the control system for smart home devices, conversation notifications and bubbles, one-time permissions, surface devices, and media in the controls.
Android 12	2021	Better notification control, Wi-fi sharing, and nearby share, privacy dashboard, approximate location, scrollable screen capture, one-handed mode, and a new UI called Material You .
Android 13	2022	Wallpaper themed of app icons, supports better haptic, notification permission, per-app languages, clipboard preview, and per-app languages.
Android 14	2023	Regional preferences, custom actions for sharing, better accessibility, better battery performance, and enhanced security.

Table 1.1: Android versions over the years

As you can see, Google continues to develop Android, adding new features and fixing problems and bugs with each new release.

History of Kotlin

Google, in 2019, made the Kotlin programming language an official language for Android app development. They made Kotlin the preferred language for app development. So, Kotlin is now the third language that can be used to develop Android apps after Java and C++.

Kotlin was developed by JetBrains in the year 2010, and it was released under the name of Project Kotlin in the year 2011. The first official version of Kotlin was released in 2016. The current version of Kotlin is 1.7, published in 2022.

The language is a cross-platform, statically typed, and general-purpose programming language that is fully interoperable with Java. This is beneficial because Java and Kotlin both run inside **Java Virtual Machine (JVM)**.

There are many benefits of choosing Kotlin over Java when doing Android App development; a few of them are listed as follows, but they will be detailed in the upcoming chapters:

- **Concise code:** As a rough estimate, code is 40% less than similar Java code.
- **Type safety:** Kotlin supports non-nullable types, which make apps less prone to the dreaded **Null Pointer Exception (NPE)** (a runtime error that crashes software/apps when a variable that is used is null) known to every Java developer.
- **Smart casting:** A feature that reduces code lines by allowing Kotlin to track conditions inside an expression.
- **Extension functions:** A Kotlin feature that helps to add functionality to a predefined class with the sub-classing technique that is used in Java language.
- **Higher-order functions:** Kotlin allows functions to accept other functions as parameters.
- **Lambdas with receivers:** Another Kotlin feature that makes code concise and makes it more readable and understandable.

So, looking at the preceding list of jargon, you might start to think that Kotlin would be hard to learn. But it is not. Instead, Kotlin is inspired by existing languages such as Java, C#, JavaScript, Scala, Groovy, and Python, and by doing so, the JetBrains developer cherry picked the best features from each language while skipping the problems seen in the other languages. Kotlin is easy to learn, and so, be it newcomers or experienced, both types of coders can learn in a matter of days. A few advanced features may take some time, but overall, Kotlin is not a complicated language. You can check the official website <https://kotlinlang.org/> to know more about Kotlin.

Market impact of mobile operating systems

The Android OS left a significant impact on the market because of its dominance as the leading mobile OS worldwide. Its open-source nature encouraged diverse device offerings and a vast app ecosystem. This spurred economic growth, global accessibility, and healthy competition. Android's integration with Google services bolstered Google's online dominance, whereas its affordability expanded internet access and digital services. Overall, Android's influence spans hardware innovation, app development, and increased digital inclusion.

When it comes to the market share worldwide of the mobile operating system, Android has been dominating the market since 2012, as the highest third-party smartphone manufacturers tend to select Android as their operating system.

The latest report from *statista.com* tells us that Google's Android is leading the market with approximately 71%, with Apple's iOS at second position with 28.3 %, and in

third place is KaiOS, a special operating system for feature phones with 0.07%. Refer to *Figure 1.1* for an illustration of the Worldwide Mobile OS Market share:

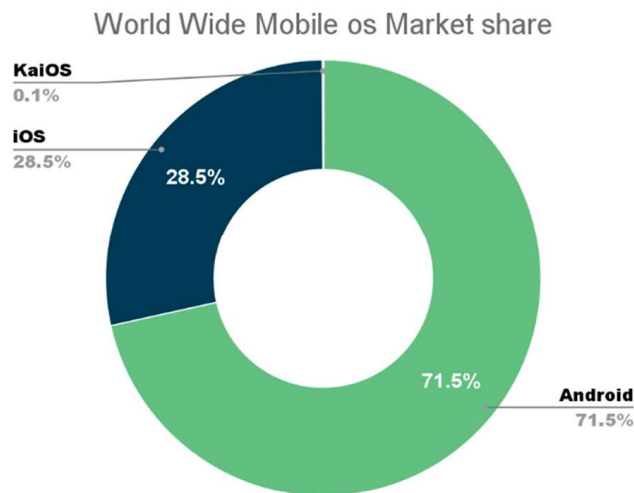


Figure 1.1: Word wide mobile OS market share

Setting up Android Studio

To begin developing Android applications on your pc, you have to install and setup only one software, which is the officially supported Android **Integrated Development Environment (IDE)** and the Android Studio. The Android Studio is developed by the JetBrains company, specifically for developing Android apps easily, and is also free to use.

It provides benefits such as follows:

- A single environment where you can develop for all Android devices.
- A fast and feature-rich emulator.
- A Gradle build system.
- Testing tools.
- Intelligent code editor.
- Integration with Play Store, and much more.

You can download Android Studio from <https://developer.android.com/studio>, Android Studio can be installed on Microsoft Windows, MacOS, Linux, and Chrome OS. You can install it in a few simple clicks. Refer to the *Figure 1.2*: