

AI-POWERED COURSE

Native Android with Kotlin & Jetpack Compose

Port a Flutter app to native Kotlin and Jetpack Compose, guided by an AI tutor.

45 modules

Kotlin

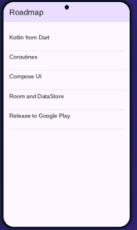
Coroutines and Flow

Capstone app

Coroutines

```
RoadmapScreen.kt

1 @Composable
2 fun RoadmapScreen() {
3     Scaffold(topBar = {
4         TopAppBar(title = { Text("Roadmap") })
5     }) { padding ->
6         LazyColumn(Modifier.padding(padding)) {
7             items(topics) { topic ->
8                 ListItem({ Text(topic) })
9             }
10        }
11    }
12 }
```



About this course

Port a working Flutter app to native Android, one concept at a time. This course teaches Kotlin from a Dart-developer's viewpoint, then rebuilds the app's interface in Jetpack Compose, its state and architecture the Android way, and its data layer with Room and DataStore. An AI tutor guides you with hints and code reviews while you write the code yourself, and you finish with a native app on Google Play.

What you will be able to do

- **Read and write Kotlin:** null safety, collections, classes, sealed types, generics, and coroutines.
- **Think in Compose:** composition, recomposition, layout, modifiers, theming, and state.
- **Build real screens:** lists, forms, navigation, adaptive layouts, and accessibility.
- **Structure an app the Android way:** ViewModel, UI state, and dependency injection.
- **Persist and sync data:** DataStore, Room, HTTP, serialisation, and the five UI states.
- **Handle real devices:** permissions, camera, media, location, notifications, and lifecycles.
- **Ship to Google Play:** logging, testing, performance, security, build variants, and release.
- **Reach beyond phones:** Kotlin Multiplatform, home-screen widgets, and legacy Views.
- **Decide with evidence:** when to choose Flutter over native, and how to justify it.
- **Work with AI professionally,** prompting for Android and reviewing AI code changes.

At a glance

MODULES

45

FORMAT

eLearning

LANGUAGE

Kotlin

PREREQUISITE

Dart/Flutter experience

PRACTICE

Porting a real app

SUPPORT

AI tutor

FINAL PROJECT

Native capstone app

Your final project

You will finish the app you ported throughout the course: a native Android build, in Kotlin and Jetpack Compose, with its own architecture, persistence, and networking, tested and configured for release, and ready to submit to Google Play.

Course roadmap

45 modules in eleven stages. Each stage builds on the one before it.

1

Why native, and the port

Modules 1–3

Why native Android, tooling and project anatomy, and planning the port from your Flutter app.

2

Kotlin from Dart

Modules 4–7

Kotlin syntax from a Dart viewpoint, null safety, collections and functional operations, classes, sealed types, and generics.

3

Coroutines and Compose basics

Modules 8–13

Coroutines, Flow and StateFlow, composition and recomposition, layout and modifiers, theming, and state and animation.

4

Building real screens

Modules 14–18

Lists and lazy layouts, forms and text input, navigation, adaptive layouts, and accessibility.

5

Architecture

Modules 19–21

Android architecture guidance, ViewModel and UI state, and dependency injection.

6

Data and networking

Modules 22–28

DataStore, Room, HTTP and serialisation, the five UI states, CRUD, authentication and secure storage, and offline-first sync.

7

Devices and platform features

Modules 29–33

Lifecycles and process death, permissions and camera, notifications and location, intents and app links, and home-screen widgets.

8

Quality and release

Modules 34–39

Logging and debugging, testing, performance, security, build variants, and release to Google Play.

9

Beyond this app

Modules 40–42

Legacy Android with Java and Views, Kotlin Multiplatform, and Flutter or native: your evidence.

10

AI and professional practice

Modules 43–44

Prompting for Android, and AI code review with controlled change.

11

Capstone

Module 45

The native app, finished.

Who this course is for

- Learners who have finished the Flutter course, or who are comfortable with Dart and Flutter.
- Flutter developers who need to ship a native Android build, or maintain one alongside their app.
- Anyone who wants to use AI to write and review Android code without accepting bad advice.

What you will need

- Android Studio and an emulator, or a physical Android device.
- A working Flutter app to port, ideally the one from the mobile course.
- Comfort with basic programming concepts; Dart experience helps but is not required.