

Android Development with Kotlin

Build, test and publish modern Android apps with Kotlin and Jetpack Compose

Mobile App Development · intermediate · Live online, instructor-led

This syllabus is a draft awaiting instructor review and may change.

A fourteen-week live course in native Android: Kotlin, Jetpack Compose, ViewModel and MVVM, coroutines and Flow, Retrofit, Room, WorkManager, Hilt, testing, security, profiling, Java interoperability and Play Store release.

Length and schedule

84 hours of live teaching, on either schedule:

- Weekday Intensive (Mon–Fri): about 17 weeks before holidays
- Weekend Evening (Sat & Sun evenings): about 21 weeks before holidays

Fees per student, whole course

- Group: Rs. 13,999
- Duo: Rs. 22,999
- One-to-One: Rs. 41,999

Launch fee — GST treatment to be confirmed.

What you will be able to do

- Write idiomatic Kotlin using null safety, data and sealed types, extension functions and lambdas
- Build responsive UIs with Jetpack Compose and manage UI state with ViewModel
- Structure an app with MVVM and unidirectional data flow, wired with Hilt
- Load data from REST APIs with coroutines and Flow, and cache it offline with Room
- Schedule reliable background work with WorkManager
- Test business logic and Compose UI, and profile performance problems
- Read and maintain Java Android code and mix Kotlin and Java safely
- Sign a release build and publish it through Play Console testing tracks

Before you start

- Basic programming in any language: variables, conditions, loops, functions and simple classes
- No previous Android or Kotlin experience required
- Comfort with Git basics is helpful; a short refresher is included
- A laptop that runs Android Studio well: 16 GB RAM recommended (8 GB minimum), a recent multi-core processor and about 20 GB of free disk space
- An Android phone for on-device testing is recommended but not required; the emulator is used in class

Tools

Android Studio, Kotlin, Java, Gradle with Kotlin DSL and version catalogs, Android Emulator and ADB, Jetpack Compose and Material 3, Jetpack Navigation, ViewModel, Room and WorkManager, Retrofit, OkHttp and Ktor client, kotlinx.serialization and kotlinx.coroutines, Hilt, JUnit, Turbine and Compose UI testing,

Syllabus

Module 1: Kotlin language essentials

The Kotlin you need for Android: safe, concise and expressive code.

1. Kotlin basics and null safety (2 h)
2. Classes, data classes and sealed types (2 h)
3. Functions, lambdas and extension functions (2 h)
4. Kotlin practice and doubt-clearing (2 h)

Module 2: Android platform fundamentals

How an Android app is built, started, paused and restored.

5. Android Studio and project structure (2 h)
6. Activities, lifecycle and intents (2 h)
7. Configuration changes and process death (2 h)
8. Platform review and doubt-clearing (2 h)

Module 3: Jetpack Compose and UI state

Declarative UI with Compose: layouts, state, lists and theming.

9. Composables, modifiers and layouts (2 h)
10. State, recomposition and state hoisting (2 h)
11. Lists, theming and side effects (2 h)
12. Compose lab and doubt-clearing (2 h)

Module 4: Navigation, ViewModel and app architecture

Move between screens and separate UI from logic with MVVM and unidirectional data flow.

13. Navigation with Navigation Compose (2 h)
14. ViewModel and UI state (2 h)
15. MVVM and unidirectional data flow (2 h)
16. Architecture review and doubt-clearing (2 h)

Module 5: Coroutines and Flow

Asynchronous work that is safe, cancellable and lifecycle-aware.

17. Coroutine fundamentals (2 h)
18. Structured concurrency, cancellation and errors (2 h)
19. Flow and StateFlow (2 h)
20. Coroutines lab and doubt-clearing (2 h)

Module 6: Networking, persistence and background work

Talk to REST APIs, store data locally and keep it in sync reliably.

21. REST APIs with Retrofit and `kotlinx.serialization` (2 h)
22. Ktor client and robust error handling (2 h)
23. Local persistence with Room (2 h)
24. Offline-first design and WorkManager (2 h)
25. Data layer lab and doubt-clearing (2 h)

Module 7: Dependency injection with Hilt and testing

Wire the app with Hilt and prove it works with unit and UI tests.

26. Dependency injection with Hilt (2 h)
27. Unit testing with JUnit and coroutine tests (2 h)
28. Compose UI tests (2 h)
29. Testing review and doubt-clearing (2 h)

Module 8: Java Android code and Kotlin–Java interoperability

Work confidently in existing codebases that mix Java, XML layouts and Kotlin.

30. Reading Java Android code (2 h)
31. Kotlin–Java interoperability (2 h)
32. Modernising a legacy app (2 h)
33. Legacy code lab and doubt-clearing (2 h)

Module 9: Security, performance and release

Protect user data, keep the app fast and ship it through Play Console.

34. Secure storage and network security (2 h)
35. Permissions and privacy (2 h)
36. Performance profiling (2 h)
37. Release builds and signing (2 h)
38. Play Console deployment and doubt-clearing (2 h)

Module 10: Capstone: offline-first library app

Plan, build and present a complete Android app from a written brief.

39. Capstone brief and architecture planning (2 h)
40. Capstone build clinic (2 h)
41. Capstone peer review (2 h)
42. Final demos and final assessment (2 h)

Final project

Build and release an offline-first Android app for a neighbourhood library. You receive a written brief and a REST API for the library's catalogue, members and loans. You deliver:- A Kotlin and Jetpack Compose app with catalogue search, book details, a member's current loans and a reservation flow, using type-safe navigation.- MVVM with unidirectional data flow: ViewModels exposing immutable UI state, repositories as the single source of truth and Hilt for dependency injection.- Networking with Retrofit or Ktor and kotlinx.serialization, with clear loading, empty and error states and retry.- Room caching so the catalogue and loans work offline, plus a WorkManager job that syncs reservations and reminds members about due dates.- Secure handling of the member's session token, a network security configuration and runtime permission requests only where needed.- Unit tests for ViewModels and repositories and Compose UI tests for the main journeys.- A small Java utility module from the brief, integrated and called from Kotlin with correct nullability.- A profiling note showing one performance issue you measured and fixed, and a signed release app bundle uploaded to an internal testing track with a draft store listing and Data safety answers. Reviewed live in the final week: you demo the app on a device or emulator, walk through the architecture and answer questions on your design choices.

Certificate

- At least 80% attendance
- All assignments submitted
- Projects passed

- Final assessment passed (60%)

A ZORK Academy course completion certificate; paying for the course does not by itself earn it.

Next step

Book a free consultation: <https://zorkacadamy.online/consultation?course=android-development>

We help you prepare for work; we do not guarantee jobs, placements or salaries.