SYNTAX COLLEGE

5-DAY NATIVE ANDROID DEVELOPMENT COURSE

Android Programming in Kotlin

From Kotlin fundamentals to production-grade Android apps with Jetpack, Compose, MVVM, Room and Coroutines.

DURATION	FORMAT	LEVEL	ANDROID
5 days · 45 hours	On-site, ~60% hands-on	Experienced programmers	Up to Android 15/16

01 About the course

This course is designed for proficient programmers with a background in Java or another object-oriented language. It opens with an in-depth exploration of Kotlin – both its functional and object-oriented sides – and then moves into the Android development environment, user-interface concepts and the core building blocks of Android apps.

The second half focuses on Android Jetpack and the MVVM architecture: ViewModel with LiveData and Flow, local persistence with Room, repositories, and asynchronous programming with Coroutines. Participants build modern UIs with Jetpack Compose alongside the classic View system, and finish with background work: services, broadcast receivers, notifications and WorkManager. Every lesson builds on the previous one around a complete project.

Who should attend

- Java, C# or C++ developers moving to Android
- Teams onboarding new Android developers
- Java-era Android teams re-skilling to Kotlin and Jetpack

What participants will be able to do

- Write idiomatic, null-safe Kotlin – functional and OOP
- Build apps with Activities, Fragments, Navigation and Compose
- Apply MVVM with ViewModel, StateFlow and Room
- Write responsive apps with Coroutines
- Run background work correctly within Android's limits

02 Day-by-day syllabus

DAY 1 Android Jetpack overview & Kotlin

Part 1 – Introduction to Android and Kotlin

- AOSP, GMS, platform architecture, native development
- AndroidX and Android Jetpack components
- Kotlin overview, variables and nullability
- Kotlin basics: if, when, loops, arrays

Part 2 – Functional programming in Kotlin

- Functions and lambdas, default arguments
- Higher-order functions
- Collection functions: filter, map, flatMap...
- Lazy sequences
- Scope, infix, inline and extension functions

DAY 2 Object-oriented Kotlin & building UIs

Part 1 – Object orientation with Kotlin

- Classes, constructors and properties
- Inheritance, interfaces and abstract classes
- Data classes; delegated properties and lazy init
- Object expressions and declarations; SAM interfaces
- Nested and inner classes; visibility and generics
- Operator overloading

Part 2 – Android Studio and user interfaces

- Android Studio project management
- Views and ViewGroups in XML layouts; event and touch handling
- Jetpack Compose fundamentals: composables, state and recomposition, Material 3
- Localization and animations
- Dialogs: alert, date/time pickers, bottom sheet

DAY 3 Android building blocks

Part 1 – Components and lifecycle

- Android app components
- Explicit and implicit Activities and Intents
- Runtime permissions with Activity Result launchers
- Activity lifecycle and DataStore

Part 2 – Single-activity architecture

- Fragments and the Fragment lifecycle

- Navigation between screens
- RecyclerView (Adapter, ViewHolder, LayoutManager) and lazy lists in Compose

DAY 4 MVVM, ViewModel, StateFlow and Room

Part 1 – Architecture

- Resources and configuration changes
- The MVVM design pattern
- Jetpack Lifecycle; ViewModel and viewModelScope
- StateFlow with or without LiveData
- SavedState module; ViewModel factories

Part 2 – Mid-project with Room

- Local (sandboxed) database with Room
- Activity results with launchers
- A full MVVM project end to end

DAY 5 Coroutines & background work

Part 1 – Asynchronous programming with Coroutines

- Responsive UI: suspension vs blocking
- CoroutineScope, CoroutineContext and dispatchers
- Builders: launch, async, withContext
- Combining suspend functions; Jetpack scopes
- The Continuation object

Part 2 – Non-UI tasks

- Notifications
- Broadcast receivers: static vs dynamic registration
- Services and foreground services
- Background execution limits
- JobScheduler and WorkManager

OPTIONAL Add-on modules (beyond the 5 days)

Dependency injection with Hilt

- DI concepts; manual vs automatic injection
- Dagger 2 and Hilt; @Inject, @Binds, @Provides
- Modules, @InstallIn and scopes; ViewModels with Hilt

Testing

- JUnit; local vs instrumented tests
- Mocks with Mockito / MockK

03 Format & logistics

Sessions	5 days × 9 hours
Hours	09:00–12:45, 13:30–16:30
Dates	Coordinated with your team
Materials	Written summaries + all source projects

Requirements

- A training room with a projector or large screen
- A laptop per participant with Android Studio installed (setup guide sent in advance)
- Android devices or emulators; USB debugging enabled
- Up to 15 participants per group

The syllabus can be shortened, extended or combined with other Syntax courses. Course materials are copyrighted and provided for participants' use.

04 Your instructor

All Syntax College courses are designed and taught personally by **Eran Katsav**, founder of Syntax College. Eran holds a B.Sc. in Computer Science from the Hebrew University of Jerusalem and an M.Sc. with honors from Tel Aviv University, and has more than 20 years of experience training engineers in industry, the defense sector and academia.

He lectures on native Android development in Kotlin and Java at Reichman University and at the Holon Institute of Technology (HIT), where he also heads the mobile lab. As an independent developer he created **Bip it**, the first Hebrew voice assistant, which reached about one million downloads in Israel; white-label versions of its engine ship in all Suzuki vehicles sold in Israel.

Selected clients: **Samsung, General Motors, Cisco, ironSource, CWT, Ophir Optronics**. Authorized supplier of the Israeli Ministry of Defense.

Bring this course to your team
Tailored syllabus, on-site at your offices, in Hebrew or English.

Eran Katsav
+972-50-6944922
info@syntax.org.il
www.syntax.org.il