Kotlin Multiplatform (KMP) - Programming Course

Led by Mr. Eran Katsav of Syntax College, this 45-hour course teaches cross-platform development using Kotlin Multiplatform (KMP), enabling you **for the first time** to build one **fully native app for all clients**: Android, iOS, Web, and Desktop (Windows, macOS, Linux) - Gain the advatnage of cross-platform (one codebase) while allowing platform specific implemtations to gain native capabilities.

About Syntax College and Mr. Eran Katsav

Syntax College is renowned for delivering a range of Android and cross-platform mobile development courses to private companies, government organizations, and the public. Clients include Samsung™, General Motors™ and Carlson Wagonlit Travel™, who have provided recommendations.

All courses are personally delivered by Mr. Eran Katsav, a senior developer and educator with a B.Sc. in Computer Science from the Hebrew University and a Master's degree with honors from Tel Aviv University. Eran teaches mobile development at Reichman University and at the Holon Institute of Technology (HIT), where he also leads the Mobile Applications & AI Lab (MAIA). With extensive real-world experience, Eran has developed and deployed live apps used by over 1 million users, including white-label solutions integrated into every Suzuki vehicle sold in Israel.

Course Overview

This course is tailored for developers with a background any object-oriented language, aiming to upskill in cross-platform development using Kotlin Multiplatform (KMP).

Participants will first gain solid grounding in the architecture, setup, and tooling of KMP and in Kotlin programmign language. Then, through hands-on exercises, they'll develop real cross-platform applications with shared code base (inluding UI and logic) across Android, iOS, and Web. The course integrates best practices from Compose Multiplatform, Ktor, Room, Coroutines, Koin/Hilt DI frameworks and more.

Why Kotlin Multiplatform (KMP)?

- **Shared Codebase for all clients: Write once, run on Android, iOS, Web And all Desktops!**
- **Native Performance and Full Platform Access**: Unlike Flutter or React Native, KMP provides direct access to native APIs and allows platform specific implantations. KMP allows you to build fully-featured apps that take advantage of the **latest platform capabilities**—including newly released Google and Apple APIs and **low-level libraries** typically inaccessible to Flutter or React Native.
- Gradual Adoption: Easily integrate into existing Android/iOS apps.
- Backed by JetBrains and Google: Ongoing development and support by the creators of Kotlin and Android OS.
- Industry Adoption: Used by Netflix, McDonald's, Forbes, Philips, VMware, Cash App and many more. Growing rapidly in global developer surveys.
- **Future-Ready and Innovative**: Built on today's most relevant system architecture — Kotlin and native platforms — to help you stay ahead of the curve with cutting-edge development practices. This is the future of Cross-Platform Development!

Course Format

Five full-day sessions (9:00–12:00, 13:00–16:00), totalling 45 hours. Includes lectures, live coding, and hands-on exercises.

Detailed Syllabus

Day 1: Kotlin Multiplatform Fundamentals and Project Architecture

- Introduction to Kotlin Multiplatform (KMP): Concept and Benefits
- Overview of KMP Targets: Android, iOS, Web, Desktop
- Setting Up Development Environment (Android Studio + KMP plugins)
- Creating Your First KMP Project and Exploring Project Structure
- Understanding Source Sets: `commonMain`, `androidMain`, `iosMain`, etc.
- Shared Code Architecture: MVI pattern (and why it suits KMP)
- **Kotlin Language Essentials: OOP + Functional style for shared code**

Day 2: Compose Multiplatform UI and expect/actual

- Understanding the `expect/actual` Mechanism
- Introduction to Compose Multiplatform UI (Android, Desktop, iOS, Web)
- Basic UI Components and Layouts in Compose MP
- Managing State in Compose MP (State Hoisting, remember)

- Navigation Patterns in Compose MP (Single activity, multiple screens)
- UI Best Practices for Shared UI
- Debugging Shared Code and Platform-Specific Issues

Day 3: Asynchronous Programming and Networking

- Kotlin Coroutines across Platforms (with structured concurrency)
- Multiplatform Networking with Ktor (client setup, logging, retries)
- Creating a Shared Repository + UseCase Layer
- JSON Parsing and Serialization with `kotlinx.serialization`
- Error Handling and Networking Best Practices
- Animations and Interactive UI with Compose MP

Day 4: Local Storage, Platform APIs, and Dependency Injection

- Dependency Injection with Koin Multiplatform
- Local Persistence Strategies: Room (experimental), SQLDelight, Realm.
- Using `expect/actual` for platform-specific code (e.g., permissions, sensors)
- Resource Management and Localization
- Image Loading with Coil (on Android/Desktop)
- Code Sharing Strategy: What to share, what not to share

Day 5: Testing, Web/JS Target, Deployment

- Writing and Running Tests in Shared Code (`kotlin.test`, MockK, Turbine)
- Introduction to Compose for Web (experimental)
- Overview of Compose for Desktop (optional or bonus)
- Publishing to App Stores:
  - Android, iOS (using Xcode), and JS/Web
- CI/CD Tools (e.g., KMMBridge, Gradle configuration for release builds)
- Future of KMP and JetBrains Roadmap
- Final Project Wrap-Up and Q&A