

**Kotlin Native**

**Course Number:** MBL-220  
**Duration:** 2 days

**Overview**

This Kotlin Native training course teaches attendees how to use Kotlin alongside native applications and libraries written in C, C++, Objective-C, and Swift. Students learn how to generate Kotlin library wrappers using headers from these other languages. Participants then learn how to write Kotlin that easily calls into native API’s and how to consume Kotlin Native libraries from other environments. Finally, attendees put it all together to see how everything fits into the wider Kotlin ecosystem by creating multi-platform projects that target the JVM, Android, JS, and Native platforms.

**Prerequisites**

Attendees must be Kotlin developers. Prior exposure to C/C++ terminology and concepts is very helpful but not essential.

**Materials**

All attendees receive comprehensive courseware.

Course outline and materials are copyrighted and owned by [Instil Software](https://instil.co/).

**Software Needed on Each Student PC**

* Windows or Mac minimum 8 GB RAM
* Android Studio installed
* Provided lab files from Accelebrate

**Objectives**

* Understand where Kotlin Native fits in the ecosystem
* Use Kotlin to create native apps, libraries, and frameworks
* Know which libraries are available on what platforms
* Be comfortable consuming C and Objective-C libraries
* Have confidence manually managing C data type memory
* Create hierarchical Gradle multi-platform projects

**Outline**

* Introduction to Kotlin Native
  + Expanding Kotlin beyond the JVM
  + Understanding the LLVM Compiler Tools
  + Platforms supported by Kotlin Native
  + JVM functionality not available in Native
  + Choosing between the CLion and IntelliJ IDEs
  + Deconstructing a simple native project
* Interoperating with C Libraries
  + Working with the provided platform libraries
  + Importing headers from external C libraries
  + Making sense of Kotlin types for native data
  + The role of CVariables, CPointers and CValues
  + Exchanging string data between Kotlin and C
  + Representing C data structures using CStructs
  + Memory management via alloc, free & memScoped
  + Invoking Kotlin callacks from C and vice versa
  + Using
* Interoperating with Objective-C / Swift
  + Working with the provided platform libraries
  + Importing headers from Objective-C / Swift
  + What Swift features are not supported
  + How Swift APIs are mapped to Kotlin
* Multithreading and Coroutines
  + The dangers of shared mutable state
  + Constraints on sharing data across threads
  + Using
  + Initializing the runtime when creating native threads
  + Utilizing Kotlin’s built-in
  + Consuming results via
  + Freezing data and
  + Breaking the rules using the
  + Transferring object ownership between threads
  + Limits on using coroutines in Kotlin Native
* Creating Multiplatform Gradle Projects
  + Choosing between the Groovy and Kotlin DSLs
  + The Kotlin Multiplatform Gradle Plugin
  + Targeting different native platforms
  + The default folder naming conventions
  + Specifying repos and project dependencies
  + Controlling compiler and linker options
  + Creating hierarchies and sub-projects
* Conclusion