

**Swift to Kotlin Android Conversion**

**Course Number:** MBL-218
**Duration:** 4 days

**Overview**

This Swift to Kotlin Android Conversion training course teaches experienced iOS/Swift developers how to seamlessly move to Android/Kotlin. Attendees learn how to build complete Android applications that conform to modern best practices and take advantage of popular frameworks, including Dagger and RxJava.

**Prerequisites**

Attendees must have several years of experience in Swift / iOS and be familiar with the core concepts of object-oriented and functional programming.

**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**

* Leverage the Java Platform
* Use Android Studio
* Convert from Swift to Kotlin
* Build user interfaces
* Test, monitor, secure, and deploy Android

**Outline**

* Introduction
* Core Concepts of the Java Platform
	+ How Kotlin emerged from Java and Scala
	+ The current JSE platform and toolset
	+ Why Google broke Android away from Oracle
	+ Java 8+ features not available on Android
	+ How Kotlin modernizes coding on Android
	+ Creating Kotlin projects in IntelliJ Community
	+ Creating mobile apps in Android Studio
* Core Tooling in JSE and Android
	+ Comparing XCode to Android Studio
	+ Managing dependencies using Gradle
	+ Pros and cons of Android emulators
	+ How Intel HAXM improves emulation speeds
	+ Configuring sample devices for testing
	+ Developer features available on devices
	+ Command-line interaction using ADB
* Making the Most of Android Studio
	+ Core functionality of Android Studio
	+ Best practices for editing Kotlin codebases
	+ Working with the Layout Inspector and Editor
	+ Configuring the SDK and AVD Manager
	+ Monitoring resource usage via the Profiler
	+ Viewing and filtering logs via LogCat
* Converting from Swift to Kotlin
	+ Minor variations in the basic syntax
	+ Mutability in Kotlin vs. Swift
	+ Similarities in support for OO and FP
	+ Limitations of class extensions in Kotlin
	+ Differences in working with collections
	+ Representing ranges in Swift and Kotlin
	+ Optionals in Swift vs. null safety in Kotlin
	+ Swift has tuples, Kotlin has
	+ Destructuring data classes and lists in Kotlin
	+ Shorthand notation for parameters in closures
	+ Comparing protocols in Swift to Kotlin interfaces
	+ Taking advantage of reflection and delegates
* Essentials of Android Applications
	+ Android has no Storyboard equivalent
	+ Understanding and editing the Manifest File
	+ Introducing Android Activities and Fragments
	+ Navigation between Activities and Fragments
	+ How a device manages the lifecycle of an Activity
	+ Designing an application around the MVVM pattern
	+ Using databinding to push data into the model
* Building Basic User Interfaces
	+ Specifying a view hierarchy as XML
	+ Reasons to avoid specifying hierarchies in code
	+ Limits of the ‘drag and drop’ approach in the IDE
	+ Using and combining the standard Android widgets
	+ The support libraries and targeting legacy versions of Android
	+ Special consideration when accepting textual input
	+ Positioning widgets by creating and nesting layouts
	+ Different options for attaching event handlers to views
	+ Using binding adapters to bind views to data sources
	+ Customizing widgets using styles and themes
	+ Creating new widgets by extending existing ones
* Enhancing the User Interface
	+ Understanding the lifecycle of an
	+ How to preserve mutable state via callbacks and bundles
	+ Starting one activity from another via intents
	+ Processing intents using filtering and receivers
	+ Creating and using files in a range of locations
	+ Advantages of Kotlin Coroutines over
	+ Using the JetBrains Kotlin Android Extensions
	+ Using the Google Android Kotlin Extensions
* Dependency Injection
	+ Using the Dagger framework for DI
	+ Understanding Compile Time Injection
	+ Configuring dependencies via
	+ Using
	+ Considerations when injecting into Activities
* Reactive Coding in Android
	+ Review of Rx and the RxJava framework
	+ Using RxKotlin for syntactic sugar
	+ Combining Rx and Kotlin Coroutines
* Accessing RESTful Services
	+ Introducing the Retrofit library
	+ Creating service clients via annotations
	+ Support for reactive streams in Retrofit
	+ Marshalling to and from JSON and XML
	+ Customizing object serialization
* Persistence in Android
	+ Supported databases on Android devices
	+ Persisting data in SQLite using Room
	+ Using annotations to specify Entity Types
	+ Using annotations to specify Data Access Objects
	+ Migrating databases between schema versions
	+ Switching databases for testing
* Security in Android
	+ Managing key pairs via the Android Keystore
	+ The security model and permissions available
	+ Prompting the user to acquire permissions
	+ Signing applications for distribution
* Background Processing
	+ Android services and their lifecycles
	+ Using the Work Manager for scheduling jobs
* Testing Android Applications
	+ TDD with JUnit, Mockito, and Hamcrest
	+ User Interface testing with Espresso
	+ Best practices for running tests in CI/CD
	+ Options for mocking RESTful services
* Deploying and Monitoring Applications
	+ Platforms for distributing beta versions
	+ Options for recording and reporting crashes
	+ Techniques for monitoring the full system
* Conclusion