

**Introduction to Kotlin**

**Course Number:** MBL-216
**Duration:** 4 days

**Overview**

This Introduction to Kotlin training course teaches attendees how to use all the programming features of Kotlin, refactor Kotlin codebases in IntelliJ, perform TDD using KotlinTest, and generate mock objects using Mockito. Participants also learn how to develop microservices via both Spring MVC and WebFlux and create basic Android applications.

**Prerequisites**

Students must be Java developers or have equivalent experience in languages like Groovy and Clojure. Knowledge of Functional Programming concepts is helpful but not essential. Students should also have prior experience with Spring Services and Mobile Development frameworks.

**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**

All students learn how to:

* Program using both the functional and object-oriented features of Kotlin
* Unit test and refactor Kotlin code via KotlinTest, Mockito and IntelliJ
* Write RESTful services in Kotlin via Spring Boot, MVC and WebFlux
* Create responsive mobile applications using Kotlin and Android Studio

**Outline**

* Introducing Kotlin on the JVM
	+ Four coding paradigms within Java 9
	+ Limits imposed by backwards compatibility
	+ The new consensus in language design
	+ A brief history of the Kotlin language
	+ Comparing Kotlin to Scala and Clojure
	+ Comparing Kotlin to Swift and TypeScript
* First Steps with Kotlin
	+ Declarations and type inference
	+ The Kotlin type system and conversions
	+ Packages, access levels and default imports
	+ Nullable types and operators for null safety
	+ Keywords for selection and iteration
	+ Options for declaring basic functions
	+ Overloading, infix functions and ranges
* Object Orientation
	+ Decompiling Kotlin classes using ‘javap’
	+ Understanding properties and backing fields
	+ A detailed explanation of primary constructors
	+ Adding extra fields and secondary constructors
	+ Extra features automatically added to data classes
	+ Overriding, abstract classes and interfaces
	+ Using object expressions as event handlers
	+ Object declarations and companion objects
* Agile Development
	+ Review of core principles of Agile development
	+ Refactoring Kotlin code within IntelliJ
	+ Performing TDD in Kotlin using KotlinTest
	+ Using Mockito to separate types from dependencies
	+ Options for BDD and Property Based Testing
* Generics and Type Parameters
	+ Revision of covariance and contravariance
	+ Common issues with bounded wildcards in Java
	+ How declaration site variance simplifies generics
	+ Support for use-site variance (aka type projection)
	+ Declaring single and multiple constraints on types
* Working with Collections Part 1
	+ Introducing the Kotlin collections library
	+ Working with mutable and immutable collections
	+ Support for destructuring types and collections
* Functional Programming
	+ Working with function references and code blocks
	+ Declaring functions as parameters and return types
	+ Using higher order functions for internal iteration
	+ Creating your own versions of ‘filter’, ‘map’ etc…
	+ Using higher order functions to prevent duplication
	+ Understanding partial invocation and currying
	+ Choosing between code blocks and local functions
	+ Common misunderstandings regarding enclosure
* Working with Collections Part 2
	+ Basic coding using ‘filter’, ‘map’ and ‘forEach’
	+ Testing against a predicate using ‘all’, ‘any’ etc…
	+ Why ‘flatMap’ is such as valuable operation in FP
	+ Distinguishing between ‘fold’, ‘foldRight’ and ‘reduce’
	+ Converting between collection types within FP
* Interoperability Between Kotlin and Java
	+ General guidelines for mixed language codebases
	+ Considerations when calling Java libraries from Kotlin
	+ Obtaining and using java.lang.Class objects in Kotlin
	+ Tips and idioms for calling Kotlin libraries from Java code
* Creating Spring Microservices in Kotlin
	+ Revision of the Spring Framework and RESTful architecture
	+ Creating Spring Boot projects in Kotlin via the Spring Initializr
	+ Declaring Spring MVC based RESTful services using annotations
	+ Injecting dependencies and configuring exception handling
	+ Creating reactive microservices using Spring WebFlux
	+ Taking advantage of the Kotlin specific routing DSL
* Creating Mobile Applications in Kotlin
	+ Revision of the Android platform and Dalvik VM
	+ Installing Android Studio and configuring emulators
	+ Introducing the sample project for tracking expenses
	+ Reverse engineering the sample activities and layouts
	+ Performing dependency injection on Android with Dagger 2
	+ Persisting application state using the Realm database
* Conclusion