

Android & Kotlin Essentials

Training Syllabus

Course Overview

If you need your team to write Android applications in Kotlin and they have little to no experience with either, this course is just what you need. In just 6.5 days, we'll have students building Android applications using Kotlin.

Who Should Take This Course

- Developers who want or need to create Android apps using the Kotlin programming language but have little to no experience with Kotlin and little to no experience with Android.
- Students should already be familiar with basic programming concepts: variables, statements, functions, arrays, data structures and common programming problems.

Syllabus

Kotlin Programming

- Understand writable & read-only properties, take a look at Kotlin's type system, and learn about nullable and non-nullable types
- Work with control flow statements as assignable expressions
- See Kotlin's scope functions — `apply`, `let`, `run`, `with`, `also`, `takeIf`, and `takeUnless` — and use them to write more concise and idiomatic Kotlin code
- Take a closer look at collections, including destructuring, type checking, smart casting, and type parameter constraints
- Use the function type to define functions that accept other functions, and learn how to optimize them by using the inlining feature of the Kotlin compiler
- Use functional programming paradigms to write complex algorithms using just a few operations, including `map`, `flatMap`, `associate`, `filter`, `zip`, and `fold`

Object Oriented Programming in Kotlin

- Apply object-oriented programming concepts in Kotlin to define your own types
- Learn about constructors, preconditions, and initialization requirements for object creation
- Inherit and override class behavior using abstract and open classes
- Define general sets of behaviors using interfaces
- Learn how to use data classes to define types that represent your application's models
- Use Kotlin's built-in singleton implementation
- Learn how to define algebraic data types with enums and sealed classes
- See how to define value classes to create more efficient classes that wrap a single property

Advanced Kotlin

- Understand type variance and generic type parameters
- Use extensions to add functionality to a type without modifying the original class
- Leverage coroutines to efficiently and safely move work between threads

Android Foundations

- Understand what an Activity is and how they are used to construct Android apps.
- Master the activity lifecycle to elegantly respond to device changes.
- Maintain UI state across rotation and process death
- Use the debugger to find and fix bugs.
- Write unit tests to verify the behavior of your app.
- Safely communicate across Activities.
- Explore the differences between versions of Android and use new features while maintaining compatibility with older versions.

Full-Featured Android

- Use Fragments to create reusable components and to respond elegantly to different types of devices.
- Display lists of items performantly with the RecyclerView API.
- Peek behind the scenes at Android's layout system and understand how to use layout tools such as ConstraintLayout.
- Customize a dialog to present modal content to the user.
- Add navigation and menu options to the Toolbar.

Beyond the Basics

- Use Room to access a database and performantly store application data.
- Use Kotlin coroutines to efficiently execute background work.
- Seamlessly communicate with other applications using Android's intent system.
- Increase the potential user base of an app by supporting multiple languages and improving the accessibility.

Connecting to the Web

- Download, parse, and present data from a web service.
- Learn more about the Toolbar on Android and how to add custom views.
- Perform background work safely in your application.
- Display content from a web site in your application.

Adding Some Fun

- Learn how to create your own custom views when the default widgets just won't do.
- Receive and respond to raw touch events.
- Understand how to increase the delightfulness of an app through animation.