Big Nerd Ranch

Kotlin Essentials

Training Syllabus

Course Overview

Discover why Google has pinned Kotlin as the preferred language for Android development with this 2-day course. You will learn Kotlin development techniques and language principles, and will walk away knowing how to migrate existing Java projects to Kotlin.

Who Should Take This Course

- Developers who are looking to explore the Kotlin language
- Developers who wish to see both object-oriented and functional programming paradigms implemented in a modern JVM language
- Developers interested in learning a language with multi-platform applications
- Android developers who want to learn new skills to advance their career
- Teams looking to upgrade their applications from Java to Kotlin

Syllabus

The Basics

- Understand writable and read-only properties and take a look at Kotlin's type system
- Work with control flow statements as assignable expressions
- Learn about defining class- and file-level functions and specifying their return types
- Leverage Kotlin's built-in numeric and string types
- Differentiate nullable and non-nullable types and how to work with null safely
- Understand safe exception handling in Kotlin

Collections and Functional Programming

- Learn how to represent series of data with mutable and read-only collection types
- Take a closer look at collections, including destructuring, type checking, smart casting, and type parameter constraints
- See how to pass functions as arguments, lambda notation, and function references
- Use functional programming paradigms to write complex algorithms using just a few operations, including map, flatMap, associate, filter, zip, and fold
- See Kotlin's scope functions — `apply`, `let`, `run`, `with`, `also`, `takeIf`, and `takeUnless` — and use them to write more concise and idiomatic Kotlin code

Objects Oriented Programming

- 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 and object declarations to create data wrappers and singletons with implementations that are automatically generated at compile time
- 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