Mobile Development Boot Camp: Kotlin / Android (Part 1): Kotlin Programming Essentials - TT4710

Master the intricacies of Kotlin to streamline mobile development, enhance code quality, and build robust, scalable applications.

Duration: 5 Days
Skill Level: Introductory
Available Format: Instructor-Led Online ; On Public Schedule

Kotlin Programming Essentials course is a five-day immersive experience geared for mobile developers eager to enhance their programming repertoire.

What You’ll Learn

Overview

A key component of our Mobile Development Boot Camp Series for Kotlin / Android, Kotlin Programming Essentials course is a five-day immersive experience geared for mobile developers eager to enhance their programming repertoire. Kotlin is the modern-day language of choice for Android development, offering readability, efficiency, and robustness. Kotlin's strong support for functional programming paradigms enables cleaner and more efficient code, especially for server-side logic and data manipulation. Through this course, you'll gain the know-how to develop sophisticated Android applications .

Throughout this interactive course, you'll spend half your time in hands-on labs and group discussions, guided by our engaging industry expert. Key focus areas include setting up your first Kotlin application, mastering variables, constants, and types, and understanding functions and null safety. The labs are designed to be a playground for your Kotlin skills, focusing on real-world mobile application scenarios. You'll delve into project-based activities like creating a basic Android app, implementing custom functions, and leveraging Kotlin's advanced features like lambda expressions and scope functions. You'll also delve into advanced subjects like lambda expressions and functional programming basics, providing you a 360-degree view of Kotlin's rich feature set.

Upon completing the course, you'll be well-versed in Kotlin's core and advanced features, from creating versatile Android apps to optimizing backend processes. You'll gain hands-on experience through labs that simulate real-world mobile application scenarios, cementing your ability to apply Kotlin in practical settings.

Objectives

This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. Throughout the course you'll learn how to:

- Start Your Kotlin Experience: By the end of our course, you'll have a solid foundation with a running Kotlin application under your belt. This sets the groundwork for all your future Kotlin-based projects.
- Master Data Essentials: We'll guide you through Kotlin's variable types and data structures, providing you with a strong understanding essential for building reliable applications.
- Sharpen Your Conditional Logic: Our focused sessions will make you comfortable with Kotlin's if/else statements and loop structures. This empowers you to craft code that can smartly adapt to different scenarios.
- Elevate Your Function Skills: The course will enable you to construct, call, and modify functions in Kotlin, leading to more modular and maintainable code.
- Explore Functional Programming: We delve into the key principles of functional programming in Kotlin. You'll learn how to work with data transformation and filtering techniques, adding another layer of efficiency to your coding skills.
- Navigate Null-Safety: Kotlin has built-in mechanisms for null-safety, and we'll show you how to utilize them. This will reduce your debugging time and make your applications more resilient.

If your team requires different topics, additional skills or a custom approach, our team will collaborate with you to adjust the course to focus on your specific learning objectives and goals.

Audience

This course is designed for experienced programmers who need to quickly get up and running with Android. Ideally students should have some experience programming in an object-oriented language such as Kotlin, Java, JavaScript or C++. This course is not for new programmers or non-technical team members.

Take Next: The following course(s) might serve as good next steps in your learning journey.

- **TT4713** Mobile Developer Camp for Kotlin / Android (Part 2): Android Development (10 days)

Pre-Requisites

This course is designed for experienced programmers who need to quickly get up and running with Android. Ideally students should have some experience programming in an object-oriented language such as Kotlin, Java, JavaScript or C++. This course is not for new programmers or non-technical team members.

Take Next: The following course(s) might serve as good next steps in your learning journey.

· TT4713 Mobile Developer Camp for Kotlin / Android (Part 2): Android Development (10 days)

Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience skill level, interests and participation.

1. Your First Kotlin Application

- Installing IntelliJ IDEA
- Your First Kotlin Project
- Creating your first Kotlin file
- Running your Kotlin file
- The Kotlin REPL

2. Variables, Constants, and Types

- Types
- Declaring a Variable
- Kotlin’s Built-In Types
- Read-Only Variables
- Type Inference
- Compile-Time Constants
- Inspecting Kotlin Bytecode

3. Conditionals

- if/else Statements
- Adding more conditions
- Nested if/else statements
- More elegant conditionals
- Ranges
- when Expressions
- when expressions with variable declarations
- when expressions without arguments

4. Functions

- Extracting Code to Functions
- Anatomy of a Function
- Function header
- Function body
- Function scope
- Calling a Function
- Writing Your Own Functions
- Default Arguments
- Single-Expression Functions
- Unit Functions
- Named Function Arguments

5. Numbers

- Numeric Types
- Integers
- Floating Point Numbers
- Formatting a Double
- Converting Between Numeric Types

6. Strings

- String Interpolation
- Raw Strings
- Reading Console Input
- Converting Strings to Numbers
- Regular Expressions
- String Manipulation
- Strings are immutable
- String Comparison

7. Null Safety and Exceptions

- Nullability
- Kotlin’s Explicit Null Type
- Compile Time vs Runtime
- Null Safety
- Exceptions
- Throwing an exception
- Handling exceptions
- Try/catch expressions
- Preconditions

8. Lambda Expressions and the Function Type

- Introducing NyetHack
- Anonymous Functions
- Lambda Expressions
- The function type
- Implicit returns
- Function arguments
- The it identifier
- Accepting multiple arguments
- Type Inference Support
- More Effective Lambdas
- Defining a Function That Accepts a Function
- Trailing lambda syntax
- Function Inlining
- Lambdas and the Kotlin Standard Library

9. Lists and Sets

- Lists
- Accessing a list's elements

- Changing a list's contents
- Iteration
- Reading a File into a List
- Destructuring
- Sets
- Creating a set
- Adding elements to a set
- while Loops
- Collection Conversion

10. Maps

- Creating a Map
- Accessing Map Values
- Adding Entries to a Map
- Modifying Map Values
- Converting Between Lists and Maps
- Iterating Through a Map

11. Functional Programming Basics

- Transforming Data
- Map; Associate
- Destructuring with functional programming
- flatMap
- Filtering Data
- Combining Data
- Why Functional Programming?
- For the More Curious: Profiling
- For the More Curious: Aggregating Data
- reduce
- fold
- sumBy

12. Scope Functions

- apply
- let
- run
- with
- also
- takeIf

- Using Scope Functions

13. Classes

- Defining a Class
- Constructing Instances
- Class Functions
- Visibility and Encapsulation
- Class Properties
- Property getters and setters
- Property visibility
- Computed properties
- Using Packages

14. Initialization

- Constructors
- Primary constructors
- Defining properties in a primary constructor
- Secondary constructors
- Default arguments
- Named arguments
- Initializer Blocks
- Initialization Order
- Delaying Initialization
- Late initialization
- Lazy initialization

15. Inheritance

- Defining the Room Class
- Creating a Subclass
- Type Checking
- The Kotlin Type Hierarchy
- Type casting
- Smart casting
- Refactoring the Tavern

16. Objects, Data Classes, and Enums

- The object Keyword
- Object declarations
- Object expressions

- Companion objects
- Nested Classes
- Data Classes
- toString
- equals and hashCode
- copy
- Destructuring declarations
- Enumerated Classes
- Operator Overloading
- Exploring the World of NyetHack

17. Interfaces and Abstract Classes

- Defining an Interface
- Implementing an Interface
- Default Implementations
- Abstract Classes
- Combat in NyetHack

18. Generics

- Defining Generic Types
- Generic Functions
- Generic Constraints
- in and out
- Adding Loot to NyetHack

19. Extensions

- Defining Extension Functions
- Defining an extension on a superclass
- Generic extension functions
- Operator extension functions
- Extension Properties
- Extensions on Nullable Types
- Extensions, Under the Hood
- Extension Visibility
- Extensions in the Kotlin Standard Library
- For the More Curious: Function Literals with Receivers
- Challenge: Frame Extension

20. Coroutines

- Blocking Calls
- Enabling Coroutines
- Coroutine Builders
- Coroutine Scopes
- Structured Concurrency
- Using an HTTP Client
- async and await
- For the More Curious: Race Conditions
- For the More Curious: Server-
- Side Kotlin
- Challenge: No Cancellations

21. Flows

- Setting Up a Flow
- MutableStateFlow
- Flow Termination
- Flow Transformations
- Error Handling in Flows
- For the More Curious: SharedFlow

22. Channels

- Dividing Work with Channels
- Sending to a Channel
- Receiving from a Channel
- Closing a Channel
- Joining Jobs
- For the More Curious: Other Channel Behaviors
- Rendezvous channels
- Buffered channels
- Unlimited channels
- Conflated channels
- Interoperation and Multiplatform Applications

23. Java Interoperability

- Interoperating with a Java Class
- Interoperability and Nullity
- Type Mapping
- Getters, Setters, and Interoperability
- Beyond Classes

- Exceptions and Interoperability
- Function Types in Java

24. Introduction to Kotlin Multiplatform

- What Is Kotlin Multiplatform?
- Planning Your Multiplatform Project
- Your First Multiplatform Project
- Defining a Kotlin/JVM Target
- Defining Shared Code
- expect and actual

25. Kotlin/Native

- Declaring a macOS Target
- Writing Native Code with Kotlin
- Launching a Kotlin/Native application
- Kotlin/Native Outputs
- For the More Curious: Kotlin Multiplatform Mobile
- For the More Curious: Other Native Platforms

26. Kotlin/JS

- Declaring Support for Kotlin/JS
- Interacting with the DOM
- The external Keyword
- Executing Raw JavaScript
- Dynamic Types
- For the More Curious: Front-End Frameworks

Related Courses

TT4710	Mobile Development Boot Camp: Kotlin / Android (Part 1): Kotlin Programming Essentials
TT4725	Swift Programming Essentials

Setup Made Simple! Learning Experience Platform (LXP)

All applicable course software, digital courseware files or course notes, labs, data sets and solutions, live coaching support channels and rich extended learning and post training resources are provided for you in our "easy access, no install required" online Learning Experience Platform (LXP), remote lab and content environment. Access periods vary by course. We'll collaborate with you to ensure your team is set up and ready to go well in advance of the class. Please inquire about set up details and options for your specific course of interest.

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.