Kotlin Programing Essentials - TT4710

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

Duration: 3 Days
Skill Level: Introductory
Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person

This hands-on course provides a solid foundation in Kotlin, a modern, concise, and powerful programming language that's rapidly becoming the standard for Android development and increasingly used across server-side, web, and multiplatform projects.

Designed for developers of all experience levels, this course explores Kotlin's syntax, type system, null safety, collections, functions, object-oriented programming features, and functional programming concepts. Through practical coding labs and guided instruction, students will gain confidence writing clean, idiomatic Kotlin code that can be applied across multiple platforms.

Whether you're transitioning from Java, building Android apps, or exploring Kotlin for backend services or multiplatform use, this course offers a strong foundation in both language features and best practices.

What You'll Learn

Overview

This hands-on course provides a solid foundation in Kotlin, a modern, concise, and powerful programming language that's rapidly becoming the standard for Android development and increasingly used across server-side, web, and multiplatform projects.

Designed for developers of all experience levels, this course explores Kotlin's syntax, type system, null safety, collections, functions, object-oriented programming features, and functional programming concepts. Through practical coding labs and guided

instruction, students will gain confidence writing clean, idiomatic Kotlin code that can be applied across multiple platforms.

Whether you're transitioning from Java, building Android apps, or exploring Kotlin for backend services or multiplatform use, this course offers a strong foundation in both language features and best practices.

Objectives

By the end of this course, participants will be able to:
- Write clean and idiomatic Kotlin code
- Use Kotlin's modern type system, including null safety and type inference
- Leverage object-oriented and functional programming features
- Work with collections, lambdas, and higher-order functions
- Handle exceptions and manage program flow with Kotlin's concise syntax
- Build reusable and modular code using classes, interfaces, and generics
- Prepare for advanced Kotlin topics like coroutines, flows, and Java interoperability

Audience

This course is ideal for:
- Developers new to Kotlin
- Java developers moving into Android or Kotlin-based projects
- Web, mobile, and backend developers interested in Kotlin's versatility
- Intermediate developers seeking a modern, expressive, and safe programming language

Pre-Requisites

- Basic programming experience in any language (e.g., Java, JavaScript, Python, etc.)
- No prior Kotlin experience required
- Familiarity with core programming concepts like variables, conditionals, and loops is helpful

Agenda

1. Your First Kotlin Application

- Your First Kotlin Project
- Creating your first Kotlin file
- Running your Kotlin file

2. Variables, Constants, and Types

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

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. Repeating code with loops

- while loops
- do-while loops
- Ranges
- For loops
- break and continue

5. Functions

- Extracting Code to Functions
- Anatomy of a Function
- Function header
- Function body
- Function scope
- Calling a Function
- Writing Your Own Functions
- The return type and the return keyword
- Default Arguments

- Single-Expression Functions
- Unit Functions
- Named Function Arguments

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

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

8. Null Safety and Exceptions
- Nullability
- Kotlin's Explicit Null Type
- Compile Time vs Runtime
- Null Safety
- Option one: checking for null values with an if statement
- Option two: the safe call operator
- Option three: the non-null assertion operator
- Exceptions
- Throwing an exception
- Handling exceptions
- Try/catch expressions
- Preconditions

9. Lambda Expressions and the Function Type
- 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

10. 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

11. Maps

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

12. Functional Programming Basics

- Transforming Data
- map
- associate
- Destructuring with functional programming
- flatMap
- Filtering Data

- filter
- Combining Data
- zip
- Why Functional Programming?
- Sequences
- reduce
- fold
- sumBy

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

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

17. Interfaces and Abstract Classes

- Defining an Interface
- Implementing an Interface
- Default Implementations
- Abstract Classes

Additional Topics (Time Permitting)

The following chapters are included for extended coverage and may be addressed as time allows. Their inclusion depends on class pacing, participant engagement, and time availability.

18. Generics

- Defining Generic Types
- Generic Functions
- Generic Constraints
- in and out

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

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

21. Flows
- Setting Up a Flow
- MutableStateFlow
- Flow Termination
- Flow Transformations
- Error Handling in Flows

22. Channels
- Dividing Work with Channels
- Sending to a Channel
- Receiving from a Channel
- Closing a Channel
- Joining Jobs
- Buffered channels
- Unlimited channels
- Conflated channels

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

Related Courses

TT4725	Swift Programming Essentials
TT4710	Kotlin Programing 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.