

Kotlin Fundamentals

Duration: 2 Days

Prerequisites:

To ensure that you have a productive and effective learning experience in the Kotlin Fundamentals course, we recommend that you meet the following minimum prerequisites:

- Basic understanding of programming concepts such as variables, control structures (if/else, loops), and functions.
- Familiarity with object-oriented programming principles, including classes and objects.
- Experience with at least one programming language, preferably Java, as Kotlin is interoperable with Java and uses similar syntax.
- Basic knowledge of using an Integrated Development Environment (IDE), such as IntelliJ IDEA or Android Studio, for writing and running code.
- Some exposure to development on the Android platform is beneficial but not strictly necessary.
- Willingness to learn and adapt to new programming paradigms and syntax.
- These prerequisites are designed to ensure that you can comfortably grasp the concepts taught in the Kotlin Fundamentals course without feeling overwhelmed. If you have any concerns about meeting these requirements, our advisors can provide guidance and resources to help prepare you for the course.

Course Description:

The Kotlin Fundamentals course is designed to introduce learners to the robust world of Kotlin programming, particularly for Android development. This comprehensive course is structured into multiple modules that cover the essentials of Kotlin, starting from setting up the development environment to mastering advanced features of the language. Learners will begin by understanding the Kotlin environment, the JVM, and how to create their first Kotlin program in Android Studio. They will then delve into the language's object-oriented features, such as classes, inheritance, and interfaces, and learn how to manage program flow and properties. The course also covers more complex topics like generics, functional programming, higher-order functions, and project structuring. By engaging in practical labs, participants will gain hands-on experience, reinforcing their learning and enabling them to apply Kotlin concepts to real-world scenarios. This course is crucial for aspiring Android developers and anyone looking to upskill in modern app development. The knowledge gained will empower learners with the skills needed to effectively use Kotlin features, such as nullability, extension functions, and exception handling, to write concise, safe, and expressive code.

Target Audience:

Koenig Solutions' Kotlin Fundamentals course is tailored for aspiring Android developers and programmers looking to master Kotlin for modern app development.

- Android Developers
- Mobile App Developers
- Software Engineers looking to switch to Kotlin
- Java Developers seeking to transition to Kotlin for Android development
- Graduate Students in Computer Science specializing in mobile app development
- Technical Project Managers overseeing Android app projects
- IT Professionals wanting to learn a new programming language
- Kotlin Enthusiasts interested in deepening their understanding of the language
- Developers looking to build cross-platform applications with Kotlin Multiplatform
- Quality Assurance Testers aiming to understand Kotlin codebases
- Tech Entrepreneurs planning to develop their own Android apps

Course Outlines:

Course Overview

- Introduction to the course
- Course Outline
- Introductions
- Introduction to Kotlin
- Kotlin Environment and the JVM
- Android Studio Environment
- Setting up an Android Studio Project
- Kotlin Hello World program
- Comments, Annotations

Lab: Hello World in Kotlin using Android Studio

Kotlin Classes

- Class definitions
- Constructors, named constructor parameters, default values
- Secondary Constructors
- Simple Properties
- Instance Creation
- Member Functions
- Function Named Parameters
- Function Default Parameters
- Function Vararg Parameters
- Functions returning Values
- Converting instances to Strings

Lab: Books and Bookshop app in Kotlin using Android Studio

REGISTER NOW!

training@trends.com.ph
(+632) 8863-2123
www.trendssacademy.com.ph

COURSE OUTLINE

Session 3: Objects, flow of control and Further Properties

- Objects
- Companion Objects
- Flow of control
- Self-reference: This
- Accessing Masked properties
- Properties and their Types
- Property Visibility
- Property Declaration Modifiers
- Custom property functions
- Nullability
- Nullable Operators
- Lazy properties
- Instance Equality
- Curly Bracket Syntax

Lab: Adding behavior to the Bookshop

Inheritance

- Inheritance in Kotlin
- Implementing Inheritance
- Rule for overriding functions
- Overriding Properties
- Rules for Polymorphic variables
- Casting and Inheritance
- The super variable
- Constructors and Inheritance

Lab: Defining Inheritance for Book class

Abstract classes and Interfaces

- Abstract Classes
- Defining an Abstract Class
- Declaring abstract functions
- Extending an Abstract Class
- Interfaces as Contracts
- Interfaces in Kotlin
- Interface Definitions
- Implementing Interfaces
- Delegation

Lab: abstract classes and interfaces

Further Classes

- Four Types of Nested Classes
- Nested classes
- Member Inner Classes
- Method Inner Classes
- Anonymous Classes / Objects
- Sealed Classes
- Inline classes
- Data Classes
- Data Classes and Destructuring Declarations
- Operators
- Infix Member Functions
- Enumeration support
- Kotlin Annotations

Lab: Implementing Data classes

Containers and Generics

- Arrays
- Collections Library
- Mutable and Immutable (read-only) collections
- Sets, Lists and Maps
- Pairs and Triples
- Collection iteration

Lab 7: Working with Containers

Functions

- What is Functional Programming?
- Kotlin as a Functional Language
- Defining Functions
- Function Objects
- Function Definitions
- Named Functions, Single Expression & Inline functions
- Anonymous Functions and Lambdas
- Callable References
- Closures
- Functions and Methods
- SAM Interfaces

Lab: Functions

Higher Order Functions and Extension Functions

- Higher Order Functions Introduction
- Functions as Parameters
- Using Typealiases for function types
- Functions as Return types

Collections & FP: foreach

- Scope functions (apply, let, also, run and with)
- Conditional functions (takeIf, takeUnless)
- Extensions
- Extension Functions and Extension properties
- Infix Extensions
- Looping over Collections
- Sorting Collections
- Grouping, Folding, Reducing and Zipping Collections

Lab: Processing containers using higher order functions

Packages and Project Structuring

- What is a Package?
- Kotlin Compiler
- Importing
- Kotlin Default Imports
- Visibility Modifiers
- Visibility Modifiers and Constructors
- Kotlin Modules

Exception Handling

- Errors & Exceptions
- Exception types in Kotlin
- Part of the Exception Hierarchy
- Exception Handling
- Kotlin and Checked Exceptions
- Try-Catch Expressions
- Finally, and returned values
- Defining New Exceptions
- • Chained Exceptions

REGISTER NOW!

training@trends.com.ph
 (+632) 8863-2123
 www.trendssacademy.com.ph