

# **Kotlin Fundamentals**

## **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

## 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 behaviour 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