



IDM TECHPARK

GUIDE'S FOR PERFECT CAREER PATHWAY



Kotlin Course Syllabus



N.S.D.C
National
Skill Development
Corporation



1

About Us

IDMTECHPARK global retail & corporate training solutions provider in Coimbatore, Erode, Trichy & Salem that offers a comprehensive range of training and placement services for both fresher's and professionals seeking new opportunities. The company commenced its IT training business in 2016. A pioneer in IT education, over the years, we have trained over 50k students. Idmtechpark has a wide range of courses, maintains education standards & provides placement assistance.

www.idmtechparkcoimbatore.com
www.idmtechparkerode.com

+91 9585305700

**2**

About IDMTECHpark Education Quality

IDMTECHPARK is managed and developed by industry specialists with more than 8 years of expertise in the field. IDMTECHPARK offers a staff of highly skilled professional trainers who deliver effective IT training in a friendly setting, concentrating on the needs of each individual to help them succeed in a demanding work world. In the book of career and success, our staff never leaves a page unturned.

www.idmtechparkcoimbatore.com
www.idmtechparkerode.com

+91 9585305700

IDMTECHPARK's versatile instructor-led training class rooms and lower-class sizes enable people to engage more easily and absorb knowledge, resulting in remarkable results for both themselves and the organizations for which they work. Our training programmes are adaptable and customizable to ensure that each participant gets the most out of their time with us. IDMTECHPARK focuses in providing hands-on IT training in over 30 different courses.

- We teach in-demand courses
- We provide impactful learning material
- Our teachers are well-selected & trained
- We follow world-class teaching methods
- Our courses include E-Projects
- We conduct technical workshops
- Exams are held and based on Exams providing Certification
- Certificates are recognized the world over
- Our course timings are flexible



3

Our Recent Placement

Idmtechpark assists students in getting job placements on successful completion of their courses. Idmtechpark also provides recruitment assistance to organizations. Idmtechpark students are shortlisted based on the organization's requirement. To make students job-ready, Idmtechpark conducts workshops e.g. How to do Group Discussions, how to behave in a Personal Interview. From time to time, job fairs & campus recruitments are conducted. Workplace skills such as time management, making effective presentations and communication skills are also provided. All this helps students find appropriate jobs in the IT industry while also helping save companies recruitment costs.

Krishnaveni M

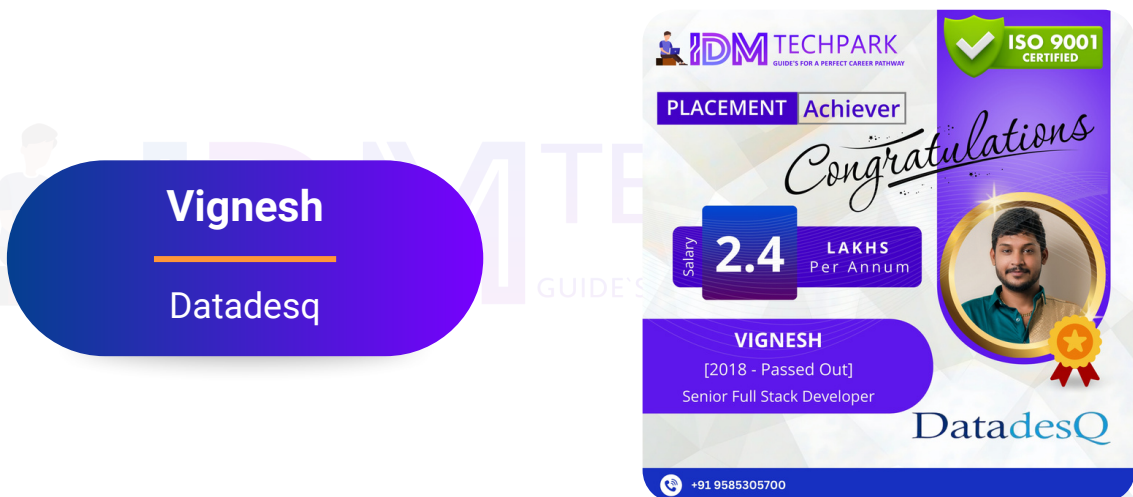
Frutterlabs





Joshwa

Xccessor



Vignesh

Datadesq



Madhumitha

Accenture

Keerthana

AJS



Brindha Boopathi

AJS



Kavin Kumar

AJS

Ajithkumar

AJS



Gowthami

Genpact



Surya

ZUCI

Pavithra

Vel Info Tech



Poovitha

Gray Matter



Ramesh

TDT

Siva Sankar
 ST Cloudspark tech



Nabeel Hisham
 VTail

Kalayarasan
 Violet Infotech



4

Our Alumni Working At



MODULE 1

Introduction to Kotlin

- Overview of Kotlin

- ☐ History and evolution of Kotlin
- ☐ Why Kotlin is preferred for Android and backend development
- ☐ Setting up Kotlin environment: IDEs, Kotlin compiler, and toolchain

- Hello World Program in Kotlin

- ☐ Writing your first Kotlin program
- ☐ Kotlin's simple syntax and structure

MODULE 2

Kotlin Basics

● Variables and Data Types

- ☐ Declaring variables: val vs var
- ☐ Primitive data types: Int, Double, Boolean, String, Char, Float, Long, etc.
- ☐ Type inference in Kotlin

● Control Flow: Conditions and Loops

- ☐ if, else, when (switch case alternative)
- ☐ Loops: for, while, do-while

MODULE 3

Functions in Kotlin

● Defining Functions

- ☐ Function syntax, parameters, and return types
- ☐ Default arguments and named arguments

● Higher-Order Functions

- ☐ Functions that take other functions as parameters
- ☐ Lambda expressions in Kotlin
- ☐ Using map, filter, and reduce

MODULE 4

Object-Oriented Programming in Kotlin

● Classes and Objects

- ☐ Defining classes and creating objects
- ☐ Constructors: Primary and secondary constructors

● Properties and Methods

- ☐ Defining and accessing properties (fields)
- ☐ Getter and setter methods
- ☐ init block and initialization

MODULE 5

Inheritance and Interfaces

● Inheritance in Kotlin

- ☐ Inheriting classes and overriding methods
- ☐ Using open and override keywords

● Interfaces in Kotlin

- ☐ Defining and implementing interfaces
- ☐ Multiple inheritance with interfaces

MODULE 6

Data Classes and Sealed Classes

● Data Classes

☐ Creating data classes and auto-generated functions (toString(), equals(), hashCode())

☐ Using data classes for immutable objects

● Sealed Classes

☐ Defining sealed classes for limited class hierarchies

☐ When to use sealed classes for better code clarity

MODULE 7

Collections in Kotlin

● Introduction to Collections

☐ Lists, Sets, and Maps in Kotlin

☐ Mutable vs Immutable collections

● Advanced Collection Operations

☐ Using filter, map, groupBy, flatMap

☐ Sorting, reversing, and merging collections

MODULE 8

Null Safety in Kotlin

● Nullable Types

- ☐ Understanding null safety in Kotlin
- ☐ Declaring nullable types with ?

● Safe Calls and the Elvis Operator

- ☐ Using ?. (safe call) and ?: (Elvis operator)
- ☐ operator to assert non-null values

● Optional Types: Optional and Option patterns

MODULE 9

Kotlin Extensions

● Extension Functions

☐ Defining extension functions to add new functionality to existing classes

☐ Extension properties in Kotlin

● Use Cases for Extensions

☐ Practical examples and common use cases in Kotlin

MODULE 10

Lambda Expressions and Anonymous Functions

● Lambda Functions

- ☐ Defining and using lambda expressions
- ☐ The concept of inline functions

● Anonymous Functions

- ☐ Defining anonymous functions and passing them as arguments
- ☐ Use of anonymous functions in callbacks

MODULE 11

Coroutines and Concurrency in Kotlin

● Introduction to Kotlin Coroutines

- ☐ Why coroutines? Introduction to asynchronous programming

- ☐ Setting up Kotlin coroutines

● Working with Coroutines

- ☐ Using launch, async, await, and delay in coroutines

- ☐ Managing background tasks and concurrency in Kotlin

MODULE 12

Error Handling and Exception Handling

● Try-Catch in Kotlin

☐ Syntax for handling exceptions with try, catch, and finally

☐ Defining custom exceptions in Kotlin

● Handling Multiple Exceptions

☐ Handling different types of exceptions

☐ Using throw keyword to throw exceptions

MODULE 13

Kotlin Standard Library

● Common Kotlin Functions

- ☐ Key functions from the Kotlin standard library (e.g., let, apply, run, with, also)
- ☐ Extension functions in the standard library

● Working with Strings

- ☐ String manipulation and formatting in Kotlin
- ☐ String templates and raw strings

MODULE 14

File I/O in Kotlin

● Reading and Writing Files

- ☐ Basic file operations: read, write, append
- ☐ Using File and BufferedReader/BufferedWriter for file manipulation

● Serialization

- ☐ Introduction to object serialization in Kotlin
- ☐ Using libraries like Kotlinx.serialization for JSON serialization

MODULE 15

Working with Dates and Times

● Handling Dates and Times

☐ Using LocalDate, LocalTime, LocalDateTime from Java's java.time package

☐ Formatting and parsing dates in Kotlin

● Kotlin's DateTime Extensions

☐ Working with time zones, durations, and periods

MODULE 16

Kotlin for Android Development

● Kotlin Basics for Android

- ☐ Introduction to Android development with Kotlin
- ☐ Setting up an Android project with Kotlin in Android Studio

● UI Components and Views

- ☐ Working with Android Views in Kotlin (TextViews, Buttons, EditTexts)
- ☐ Event handling in Android using Kotlin (click listeners, touch events)

MODULE 17

Kotlin Multiplatform

● Introduction to Kotlin Multiplatform

- ☐ What is Kotlin Multiplatform (KMP)?
- ☐ Sharing code between platforms (Android, iOS, Web, and Backend)

● Setting Up KMP Projects

- ☐ Creating a Kotlin Multiplatform project
- ☐ writing shared logic in Kotlin for multiple platforms

MODULE 18

Working with Dependency Injection in Kotlin

● Introduction to Dependency Injection

- Why use DI in Kotlin and its advantages
- Manual DI in Kotlin using constructor injection

● Using DI Frameworks in Kotlin

- Introduction to DI frameworks like Dagger, Koin, and Kodein in Kotlin
- Setting up Koin for dependency injection in Android applications

MODULE 19

Testing in Kotlin

● Unit Testing with Kotlin

- ☐ Writing unit tests with the JUnit framework
- ☐ Using Mockito for mocking dependencies in Kotlin

● UI Testing in Kotlin

- ☐ Setting up Espresso for UI testing in Android with Kotlin
- ☐ Writing test cases for user interactions

MODULE 20

Reflection in Kotlin

● What is Reflection?

☐ Using Kotlin reflection to inspect and modify class metadata

☐ KClass, KProperty, and KFunction

● Use Cases for Reflection

☐ Dynamically invoking methods and accessing properties at runtime

☐ Reflection-based libraries and frameworks

MODULE 21

Working with Annotations in Kotlin

● Using Annotations

- ☐ Creating and using custom annotations in Kotlin
- ☐ Meta-annotations in Kotlin

● Annotation Processing

- ☐ Annotation processing at compile-time with Kotlin
- ☐ Introduction to frameworks using annotations (e.g., Dagger)

MODULE 22

Kotlin for Backend Development

● Kotlin for Web Development

- ☐ Setting up a Kotlin backend with Ktor or Spring Boot
- ☐ Building RESTful APIs using Ktor or Spring Boot

● Database Integration

- ☐ Working with databases using Kotlin (e.g., Exposed ORM, JDBC)
- ☐ CRUD operations with databases in Kotlin backend

MODULE 23

Kotlin DSL (Domain-Specific Languages)

● Introduction to Kotlin DSL

- ☐ What is a Domain-Specific Language?
- ☐ Why Kotlin is suited for DSL creation

● Creating a DSL in Kotlin

- ☐ Implementing simple DSLs in Kotlin (e.g., UI DSL, Build DSL)
- ☐ Examples of popular Kotlin-based DSLs

MODULE 24

Kotlin Performance Optimization

● Optimizing Kotlin Code

☐ Best practices for writing efficient and performant Kotlin code

☐ Memory management in Kotlin

● Profiling Kotlin Applications

☐ Using profiling tools like Android Profiler, VisualVM, and others to detect performance issues

MODULE 25

Capstone Project

● Capstone Project

☐ Building a full-fledged Kotlin application (mobile, backend, or multiplatform)

☐ Incorporating all learned topics: state management, coroutines, UI/UX, and testing

● Code Review and Best Practices

☐ Reviewing code for optimization, readability, and maintainability

☐ Final project presentation and feedback

Thank You

+91 9585305700

www.idmtechparkcoimbatore.com

www.idmtechparkerode.com