

Mobile Development

One codebase, both app stores. You learn Dart, build screens that feel right on each platform, use the camera and location, and get a signed release onto a real phone.

Foundational

4 phases

20 sessions

30 contact hours

Syllabus

4 phases, 20 sessions. Each phase ends in something you have built.

1

Dart Before the Widgets

Learn the language and its type system before the widget tree hides it.

5 sessions

\$72 alone

1.1 Setup That Actually Finishes

Run flutter doctor to green, boot an emulator or simulator, and know what to do when the Android toolchain refuses.

[FREE PREVIEW](#)

1.2 Dart, Typed and Null-Safe

Declare variables and functions with real types, and explain why String? is a different type from String.

1.3 Control Flow and Collections

Use lists, maps, loops, and switch expressions with pattern matching instead of a stack of if statements.

1.4 Classes, Records and Sealed States

Write immutable models, parse JSON into them, and model loading, data, and error as types you cannot forget to handle.

1.5 Futures, async/await and Errors

Await work that takes time, catch what it throws, and decide what the user sees during every one of those moments.

By the end: **A command-line Dart program with a tested model layer that loads, fails, and recovers.**

2

The Widget Tree

Build screens that lay out correctly, accept input, and navigate.

5 sessions

\$72 alone

2.1 Everything Is a Widget

Compose MaterialApp, Scaffold, and AppBar, and rebuild a tree instead of mutating a view.

2.2 Constraints Go Down, Sizes Go Up

Lay out Row, Column, Expanded, and Padding, and diagnose the yellow overflow stripes yourself.

2.3 setState and the Smallest Rebuild

Hold state in a StatefulWidget and push it down to the smallest widget that actually needs it.

2.4 Lists, Forms and Input

Build a scrolling ListView.builder, validate a Form with controllers, and design the empty state you forgot.

2.5 Navigation with go_router

Declare routes, pass arguments, handle deep links, and see why imperative Navigator calls stop scaling.

By the end: **A multi-screen app with lists, forms, and a working back stack, running on a real phone.**

3

State, Data and Architecture

Get logic out of widgets and data out of memory.

6 sessions

\$72 alone

3.1 ChangeNotifier and ListenableBuilder

Share state across screens using the mechanism Flutter ships with, before reaching for any package.

3.2 Repositories and ViewModels

Split a feature into UI, ViewModel, and data layers following Flutter's own MVVM guidance.

3.3 Riverpod for Real Apps

Wire providers for async state and dependency injection, and override them in a test with no device attached.

3.4 Talking to an API

Fetch JSON over http, decode it into typed models, and handle timeouts, empty results, and a server that returns 500.

3.5 Local Persistence

Store settings with SharedPreferencesAsync, store records in Drift, and write a schema migration on purpose.

3.6 Tests That Are Worth Writing

Unit test the ViewModel, widget test a screen, and use DevTools to find the rebuild you did not intend.

By the end: **An offline-first app with a repository, a ViewModel under test, and records that survive a restart.**

4

The Device and the Stores

Use one real device capability and get a build past both stores' 2026 gates.

4 sessions

\$72 alone

4.1 One Capability, Done Properly

Request camera or location permission, handle the denied and permanently-denied paths, and open system settings when you must.

4.2 Android Build Reality

Set `compileSdk` and `targetSdk` in the Gradle Kotlin DSL, meet the 16 KB page size rule, and produce a signed release bundle.

4.3 iOS Build Reality

Build against the iOS 26 SDK in Xcode 26, resolve dependencies with Swift Package Manager, sign, and upload to TestFlight.

4.4 Store Submission, Calendar Included

Prepare icons, screenshots, and the data safety and privacy forms, and plan Play's 12 testers across 14 continuous days.

By the end: **A signed Android App Bundle in closed testing plus a TestFlight build of the same app.**

Tools you will use

Flutter 3.44

Dart 3.12

material_ui / cupertino_ui

Riverpod 3

go_router

Drift

Flutter DevTools

Xcode 26 + Android Studio

What you will build

1 Multi-Screen Notes App

Widgets, forms, and a back stack that behaves

2 Offline-First Habit Tracker

A Drift database, a repository, and a ViewModel under test

3 Shipped Build

Signed App Bundle in closed testing plus a TestFlight invite

Registration

Phase 1: Dart Before the Widgets	\$72
Phase 2: The Widget Tree	\$72
Phase 3: State, Data and Architecture	\$72
Phase 4: The Device and the Stores	\$72
Whole track, all 4 phases	\$240

Buying every phase separately costs \$288, so the whole track saves \$48.

No payment is taken online. Registering creates an order and payment instructions are sent with it. Access opens once payment is confirmed.