

CET215: Mobile Application Development

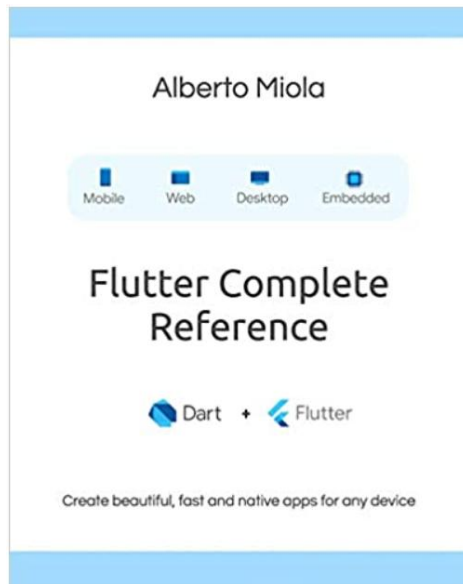
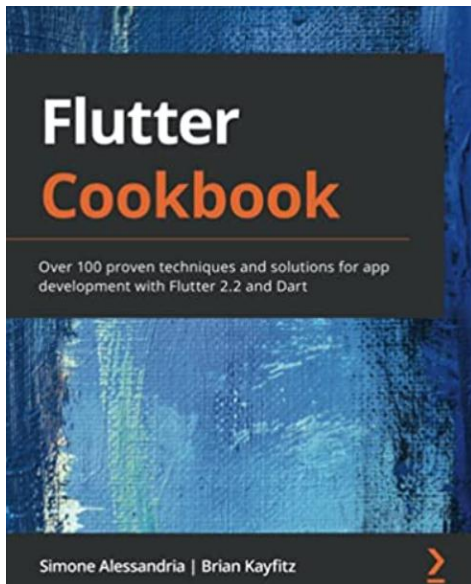
Lecture 1: Overview





References

- **Flutter Complete Reference:** Create beautiful, fast and native apps for any device
- **Flutter Cookbook**





LMS

- Course Materials: Lectures and Lab Assignments
- Online Quizzes
- Project Submission
- Lab Assignment Submission





Grades

- Midterm: 15 Degree
- Activities
 - Project: 20 Degree
 - Quizzes (Practical/LMS): 20 Degree (6 quizzes, Inside lab, best 5)
 - Practical Assignments: 20 Degree (LMS + Discussion)
- Final Exam: 25 Degree





Introduction

- Mobile application development:
 - Creating software applications that run on mobile devices (smartphones, tablets).
- Existing Operating Systems:
 - iOS
 - Developed by Apple
 - iPhone, iPad
 - Android
 - Developed by Google
- Programming Languages for Each OS: (Native Languages)

مطلی

OPERATING SYSTEM	PROGRAMMING LANGUAGES
iOS	Swift, Objective-C
Android	Kotlin, Java





Available Technologies for Mobile App Development



Native Development:

- Java, Kotlin for Android
- Objective-C, Swift for IOS



Hybrid Development:

- Uses web technologies (HTML, CSS, JavaScript) wrapped in a native container
- Use internal WebView to run the APP + Native plugins
- Ex: Ionic, Cordova



Cross-Platform Development:

- Single codebase for multiple platforms
- Compiled to native components
- Ex: Flutter, React Native, Xamarin, .NET MAUI





Native VS Hybrid VS Cross-Platform

FEATURE	NATIVE	HYBRID	CROSS-PLATFORM
Performance	High	Moderate	High
Development Time	Long	Short	Moderate
Code Reusability	Low	High	High
Platform Support	Single platform	Web-based	Multi-platform
App Size	Small, optimized	Larger, includes extras	Often larger, native components
Examples	Swift, Kotlin	Ionic, Cordova	Flutter, React Native





Cross-Platform Trending Technologies



React Native:

- Best for web developers who know JavaScript
- Apps feel more "native" but can be slower



Flutter:

- Great for custom designs (beautiful design)
- Faster apps
- Uses Dart, which is easy to learn





What is Flutter

It's a **framework** for building user interfaces with **Dart**

Framework

A collection of packages & utility functions you may use in your code

Dart



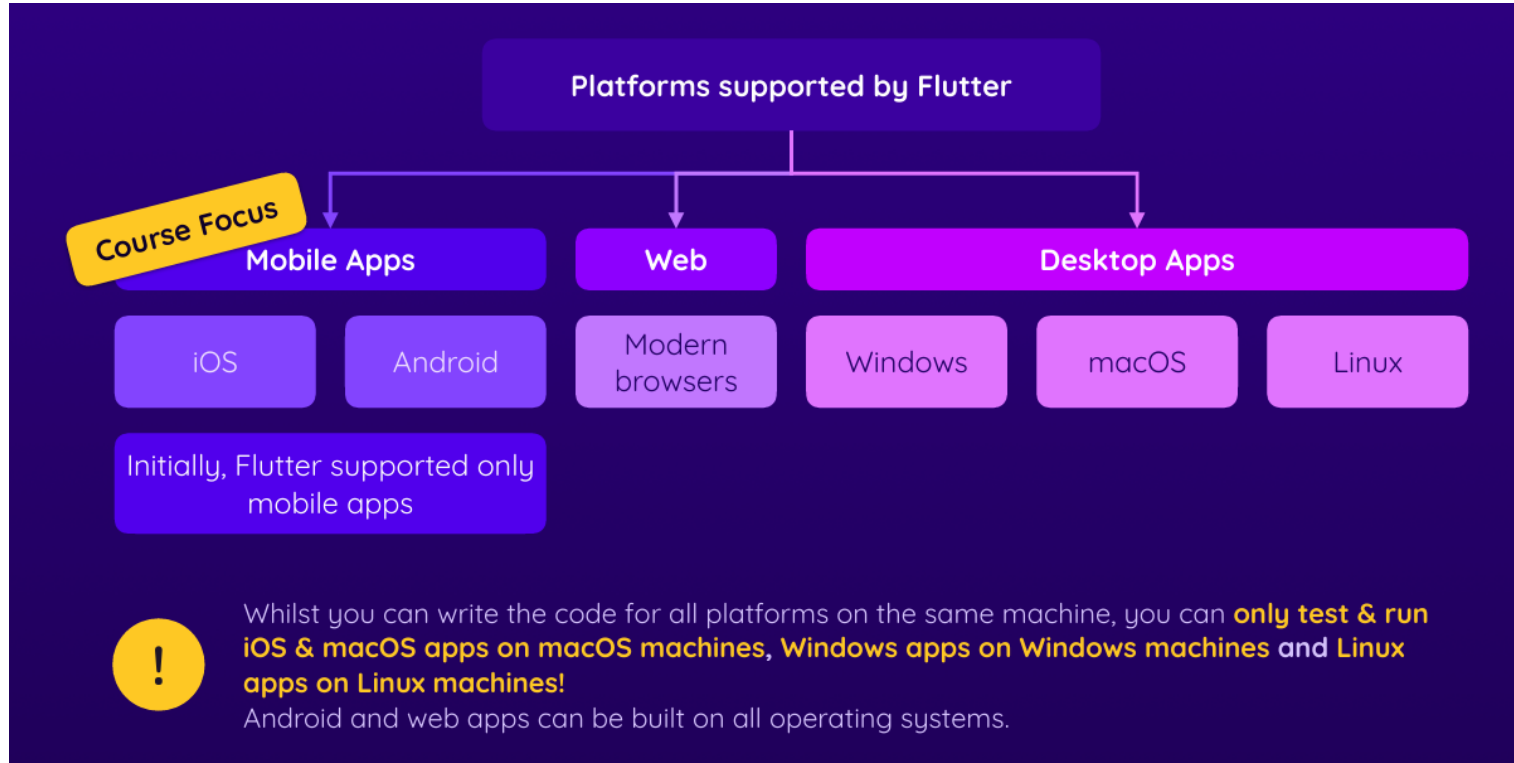
A programming language developed by Google

Main usage: Flutter app development





What I need to install?





What I need to install?

1



Flutter SDK

Flutter SDK

For managing Flutter projects

Git

Version control software, used internally by Flutter SDK

2



Platform Tools

Android Studio

Used by Flutter SDK & needed for Android app deployment

XCode

Used by Flutter SDK & needed for iOS app deployment

3



Virtual Devices

Android

Preview Flutter apps on virtual Android devices

iOS

Preview Flutter apps on virtual iOS devices





How Flutter Work?

Flutter Widgets

In Flutter, a widget is a description of part of a user interface. It is a building block for creating the visual elements of an app, such as buttons, text, images, and layouts.





Widgets Examples

Container

A widget that provides a rectangular visual element for displaying other widgets.



Image

A widget that displays an image on the screen



App Bar

A widget that provides a top app bar with navigation buttons and a title.



Text

A widget that displays a string of text on the screen.



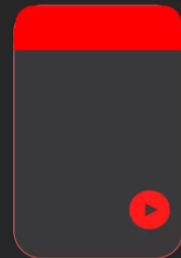
List View

A widget that displays a scrollable list of widgets.



Floating Action Button

A widget that provides a floating action button for triggering a primary action in the app.





Widgets Can be Styled

Style Properties

color

Padding

Margin

Border

Background
image

Box Shadow

Opacity

Gradient





What Next?

- 1 or 2 lectures: Overview on Dart Language + OOP
- Remaining Lectures: Flutter + APPs

