



MOBILE APPLICATION DEVELOPMENT

ANDROID (2017)

LECTURE 01: INTRODUCTION

COURSE OBJECTIVES

- ▶ Provide a foundational understanding of mobile application development:
 - ▶ Describe common problems and their solutions.
 - ▶ Present standard design patterns and best practices.
 - ▶ Familiarize students with tools and environments common to the industry.
- ▶ Demonstrate practical implementations of the above concepts:
 - ▶ Use Android as an example development platform.
 - ▶ Use Kotlin as an example development language.

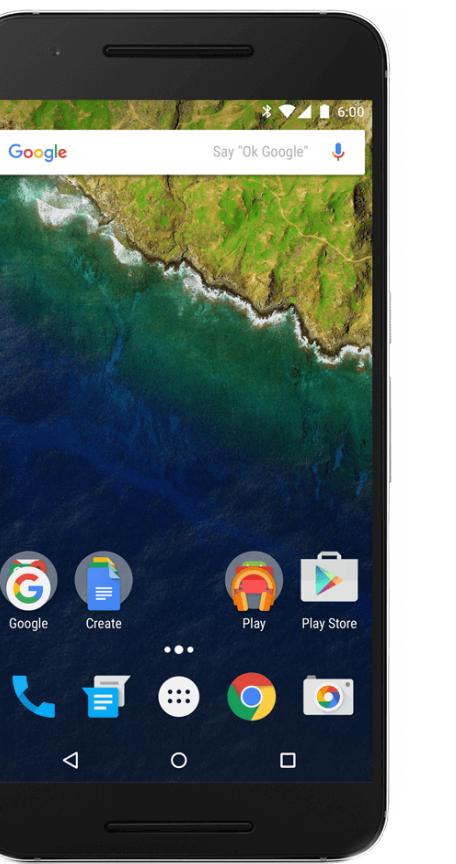
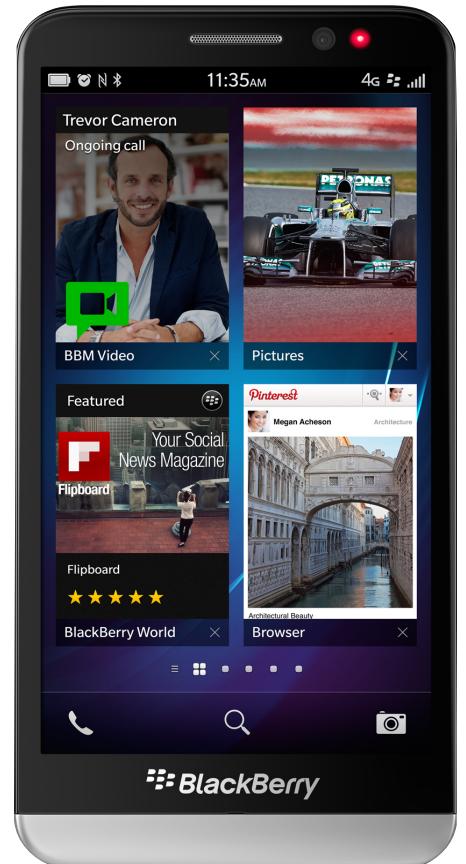


ABOUT ME

- ▶ Professional iOS / Android / C Developer
- ▶ Computer Science / Psychology Graduate
- ▶ Former Professional Photographer
- ▶ Video Game Enthusiast



THE MOBILE LANDSCAPE



THE MOBILE LANDSCAPE



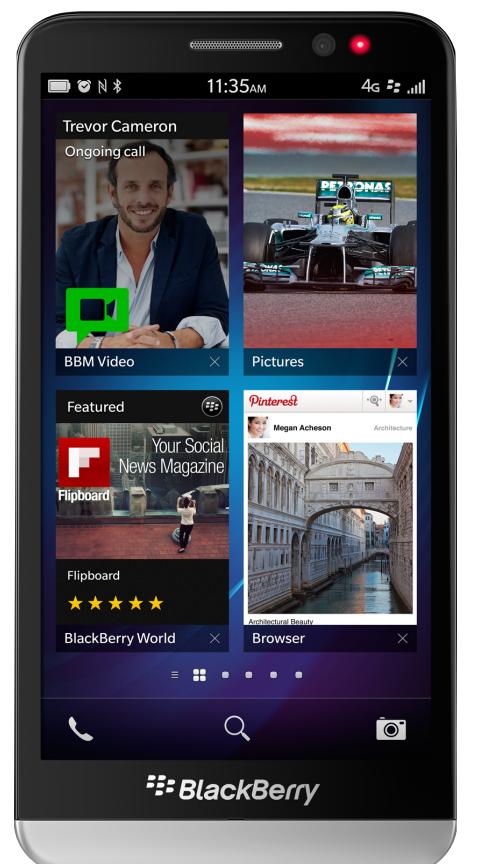
Nokia N8



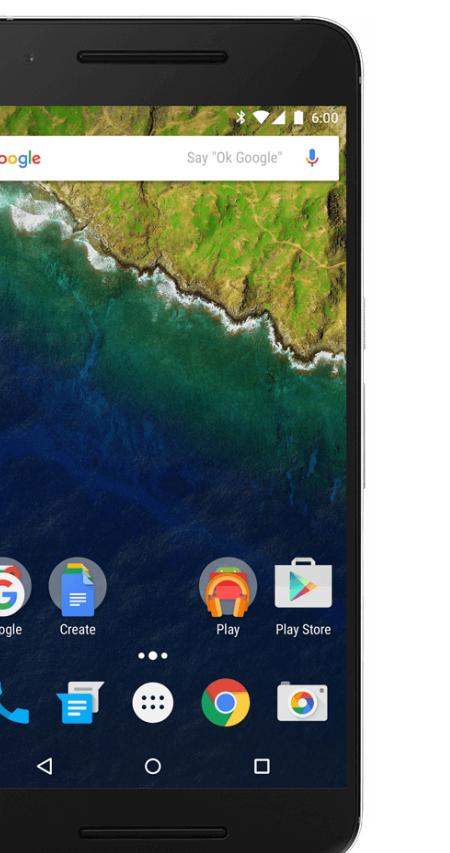
iPhone 6+



Lumia 950



BlackBerry Z30



Nexus 6P



Palm Pre 2

THE MOBILE LANDSCAPE



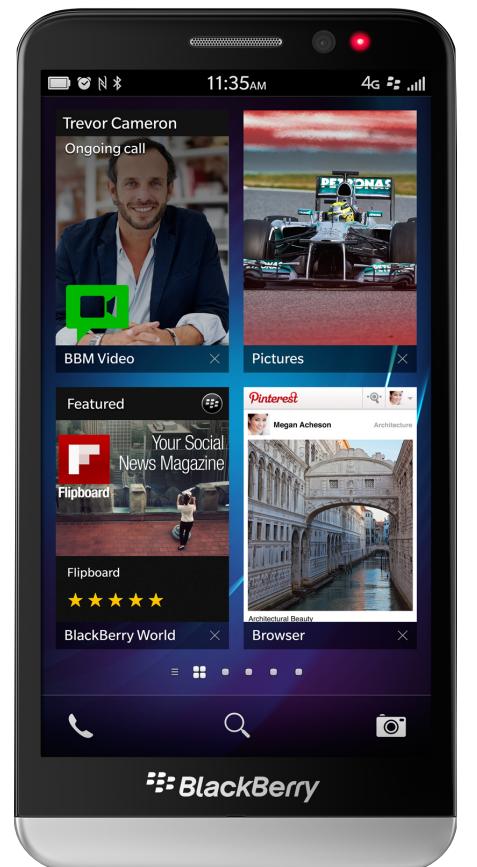
Symbian



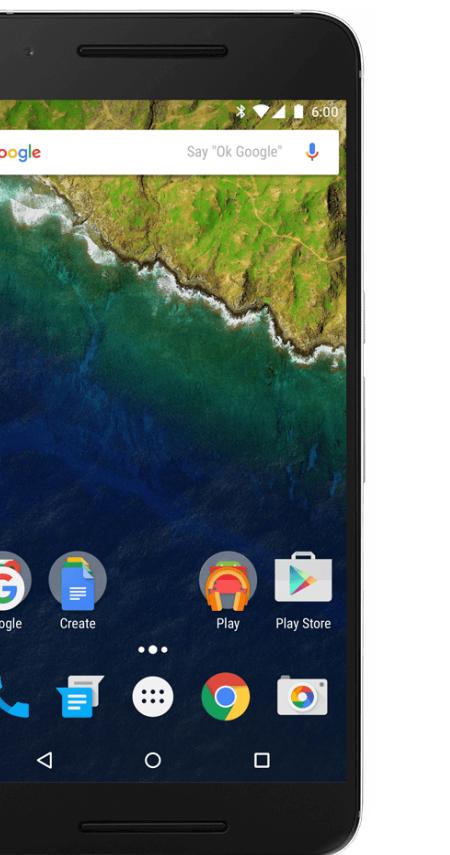
iOS



Windows



BlackBerry OS



Android



WebOS

THE MOBILE LANDSCAPE



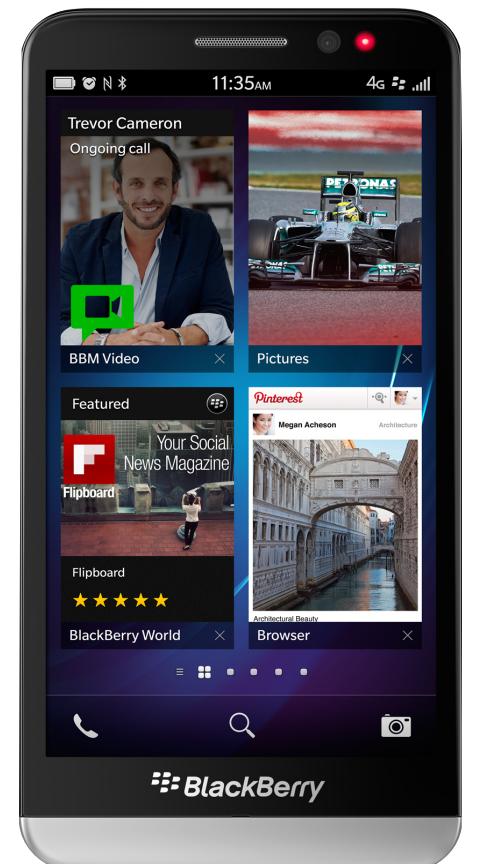
C++/Java/WRT



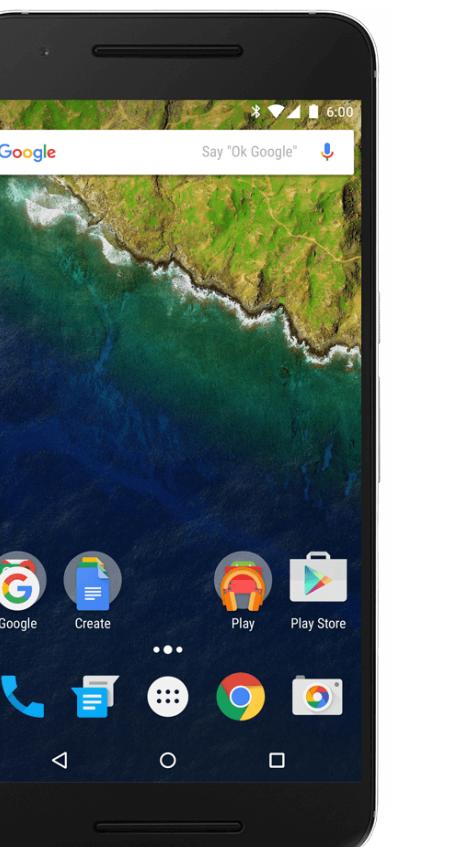
Swift



C#



Java

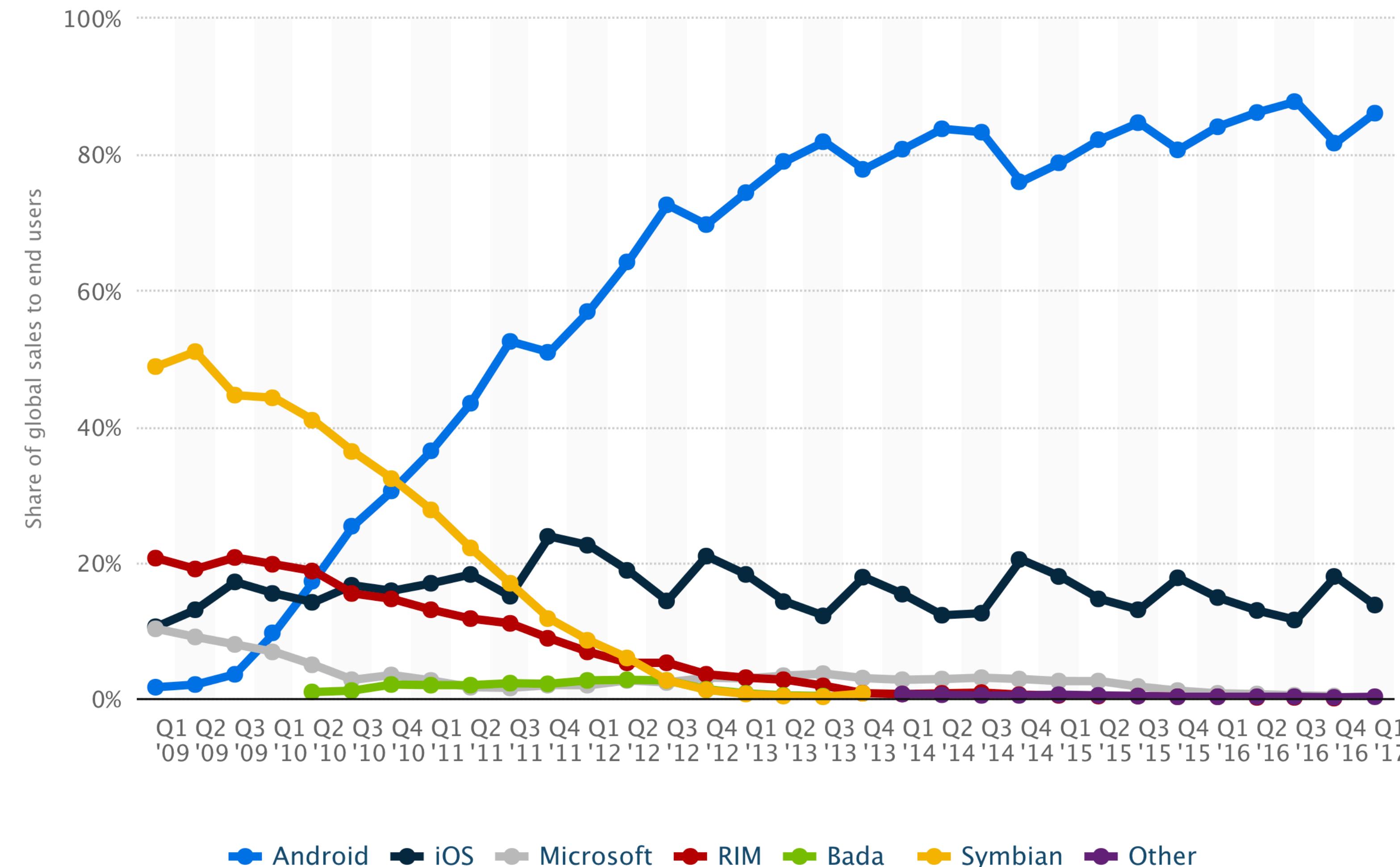


Java

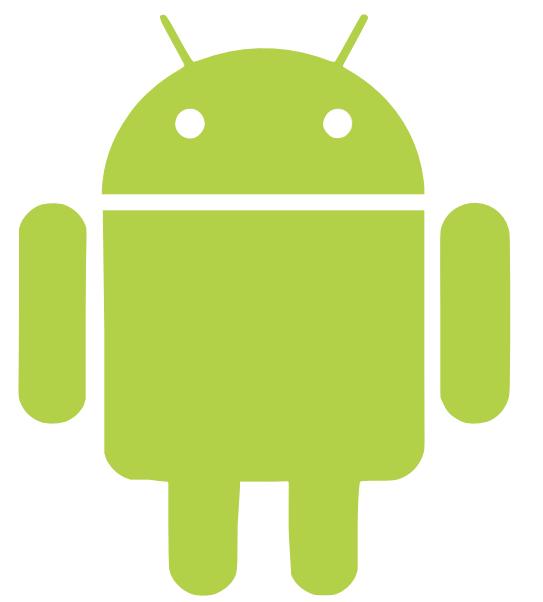


JavaScript

THE MOBILE LANDSCAPE

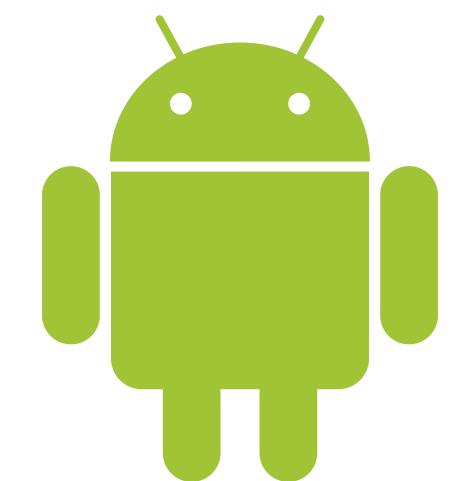


WELCOME TO ANDROID



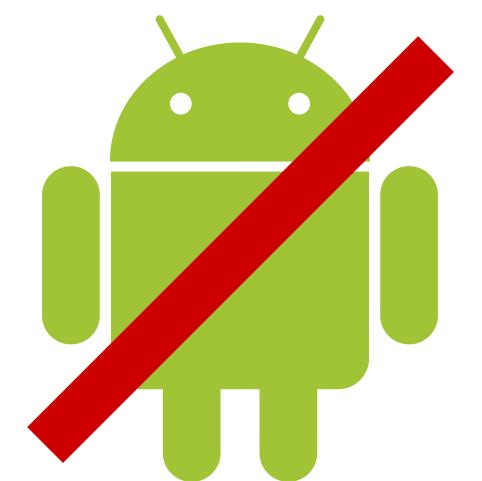
WHY ANDROID?

- ▶ It is the most popular mobile OS in the world, and likely will be for many years.
- ▶ You can develop for Android on virtually any platform, and test on many devices.
- ▶ Its Java / JVM base means that a massive software ecosystem is available.
- ▶ The platform's popularity means a vibrant community of programmers exists.
- ▶ Android arguably has some of the best tooling and SDKs of any mobile platform.
- ▶ Android has few restrictions on the kinds of software you can create.



WHY NOT ANDROID?

- ▶ Adoption of new versions of Android has been and will be extremely slow.
- ▶ Fragmentation of devices is still a problem, and likely always will be.
- ▶ A relatively small number of Android users actually purchase software.
- ▶ Hardware running Android is often vastly inferior to the competition.
- ▶ Google's SDK / library situation on Android is confusing and frustrating.
- ▶ Google's application of their own design guidelines is inconsistent.



ANDROID STUDIO

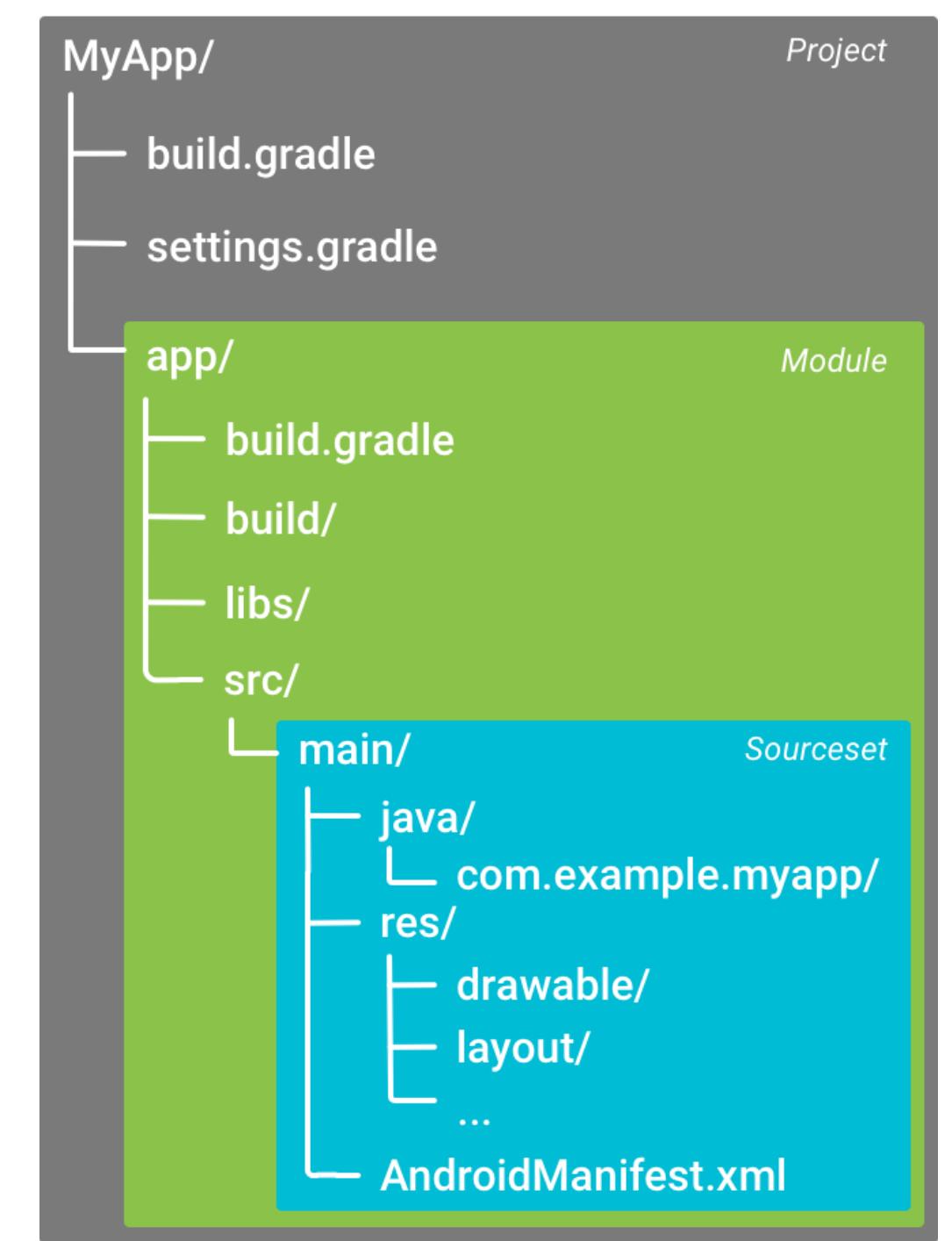
- ▶ Integrated Development Environment Based On IntelliJ
- ▶ Source Code Editor With Advanced Features
- ▶ Compiler / Debugger / Build Automation
- ▶ Graphical, XML-Based Interface Builder
- ▶ Android Emulator
- ▶ Plugin System



ANDROID PROJECT STRUCTURE

► Projects and Modules

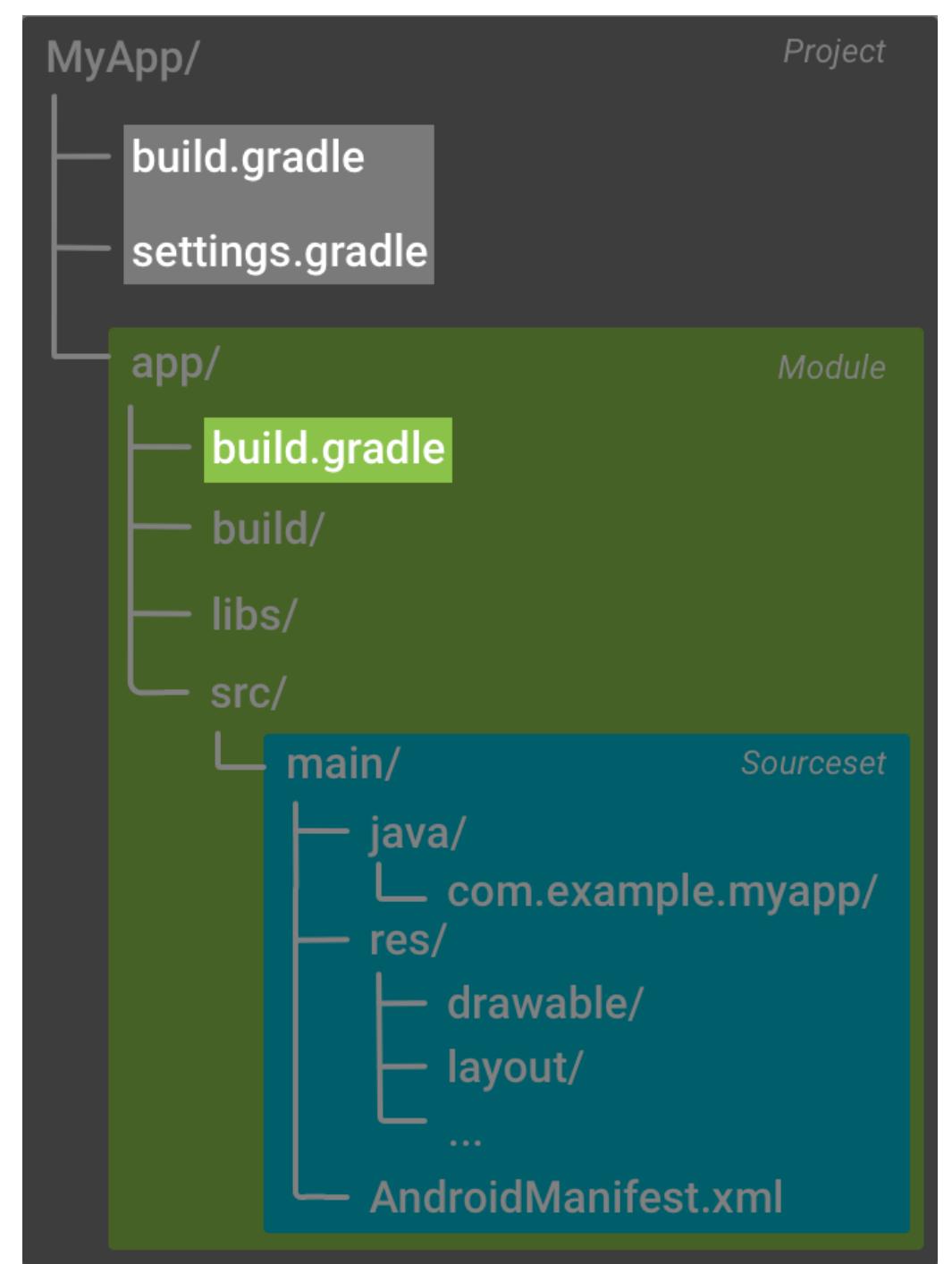
- A project in Android is a collection of modules.
- A module is something like a library or application.
- An Android project may contain only one module, or many.
- Each module contains the code and libraries it needs to be built.
- Modules in projects may depend on one another, useful for modularizing code.



ANDROID PROJECT STRUCTURE

▶ Gradle Files

- ▶ Specify build dependencies for projects and modules.
- ▶ Describe configurations for the Gradle build system itself.
- ▶ Let the programmer specify different application versions.
- ▶ Provide the ability to obfuscate and minify application code during builds.
- ▶ Allow programmers to divide large projects into multiple connected modules.



ANDROID PROJECT STRUCTURE

► AndroidManifest.xml

- ▶ Identifies an application to the system.
- ▶ Enumerates the permissions the application requires.
- ▶ Specifies the minimum API level required by the application.
- ▶ Enumerates any API libraries (besides the default) the application links with.
- ▶ Declares hardware / software components that the app uses on the device.

