

EE5415

Mobile Applications Design and Implementation

Course Overview



Agenda

- Lecture Session
 - **Location: Li Building, Room 1601**
 - Introduction to Android
 - Course Overview
 - Forming Project Groups
- Lab Session
 - **Locations: Yeung Building, Room P1406 and Room P1442**
 - A list of lab-assigned students can be found on the course website.
 - Lab01 : Android Development System

Session I

- Teaching staff
- Introduction of Android
- EE5415 resources
 - Course website
 - Grading: Labs, Test, Mid-Term Exam, Projects and Final Exam
 - Group Project arrangement

Teaching Staff

- **Course Instructor**
 - **Dr. Lai-Man Po**
 - eelmpo@cityu.edu.hk
 - Room G6506, Green Zone, 6/F, AC1
 - Phone: 3443-7779
- **Technician**
 - **Mr. C. Y. CHENG**
 - cycheng@ee.cityu.edu.hk
 - Student Terminal Room (P1442)
- **Graders**
 - **Mr. ZHAO Yuzhi** (yzzhao2-c@my.cityu.edu.hk)
 - **Mr. XIAN Pengfei** (xian.pf@my.cityu.edu.hk)
 - Room P1412, EE Lab, Purple Zone, 1/F, AC1

Course Website

- Course Website:
 - <http://www.ee.cityu.edu.hk/~lmpo/ee5415/index.html>
 - Students should check the course website regularly for announcements about the course.
 - Copies of the lecture, assignment, project information are all available in course website with various formats.
- Messages, Schedule, Projects, Links
 - Username: **students**
 - Password: **android10**

Course Description

- This course aims to provide students with **hands-on experience** on design and development of Android Apps.
- Students will learn **skills** leading to **creating and deploying** Android apps, with particular emphasis on software engineering topics including software architecture, software process, usability, and deployment.

What is Android?

What is Android?

Simple Answer:

- **Android is an operating system for smartphones**
 - Based on Linux
 - Supports mobile devices (phones, tablets, watches, etc)

Better Answer:

- **Android is a software stack, which consists of**
 - Operating system
 - Middleware
 - Key applications

Android Supports Different Devices

- Mobile devices – core framework
 - Phones, Tablets, and Watches



- Other form factors
 - Wearable, autos, and TV



The Android SDK (Software Development Kit)

- **Compiler**
 - Build and package apps for distribution
- **Debugging tools**
 - Test and refine your apps
- **Development tools**
 - **Android Studio:** A complete IDE

Languages and Tools

- Programming Languages for Android Development
 - **Java** (used in this course)
 - **Kotlin** (supplemental materials are provided)
- Configuration files especially UI Layout file
 - **XML Markup Language**
- Developer Tools
 - **Android Studio**
 - Integrated development environment (IDE) based on IntelliJ IDEA
 - Free and can be downloaded from Google

Android's Beginnings

- Android first released in 2008
 - Only worked on cell phones
- Support for tablets added in 2011
- Other features added in the last few years

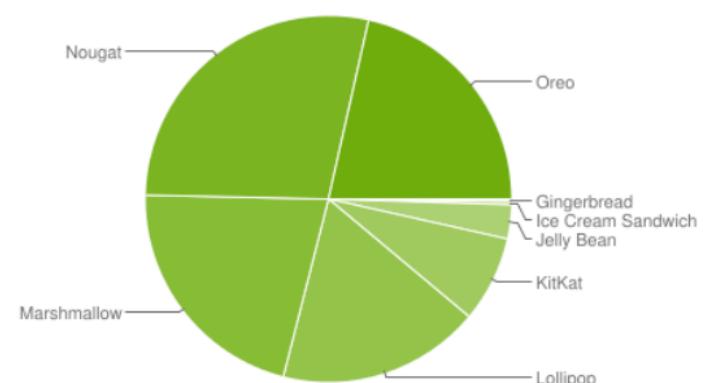


https://en.wikipedia.org/wiki/Android_version_history

Android Versions

- Android 1.0 (Alpha) API 1 (September 23, 2008)
- Android 1.1 (Beta) API 2 (February 9, 2009)
- Android 1.5 (Cupcake) API 3 (April 27, 2009)
- Android 1.6 (Donut) API 4 (September 15, 2009)
- Android 2.0-2.1 (Eclair) API 5 – 7 (October 26, 2009)
- Android 2.2 (Froyo) API 8 (May 20, 2010)
- Android 2.3.3 – 2.3.7 (Gingerbread) API 9 – 10 (December 6, 2010)
- Android 3.0 – 3.2.6 (Honeycomb) API 11 -13 (February 22, 2011)
- Android 4.0 – 4.0.4 (Ice Cream Sandwich) API 14 -15 (October 18, 2011)
- **Android 4.1 – 4.3.1 (Jelly Bean) API 16-18 (July 9, 2012)**
- Android 4.4 (KitKat) API 19-20 (October 31, 2013)
- **Android 5.0-5.1.1 (Lollipop) API 21-22 (November 12, 2014)**
- Android 6.0-6.0.1 (Marshmallow) API 23 (October 5, 2015)
- Android 7.0-7.1.1 (Nougat) API 24-25 (August 22, 2016)
- Android 8.0-8.1 (Oreo) API 26-27 (August 21, 2017)
- Android 9.0 (Pie) API 28 (August 6, 2018)
- **Android 10 (Q) API 29 (September 2019)**

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.1%
4.2.x		17	1.5%
4.3		18	0.4%
4.4	KitKat	19	7.6%
5.0	Lollipop	21	3.5%
5.1		22	14.4%
6.0	Marshmallow	23	21.3%
7.0	Nougat	24	18.1%
7.1		25	10.1%
8.0	Oreo	26	14.0%
8.1		27	7.5%



The History of Android (1)

- Version 2.0 (Éclair, October 2009)
 - Multiple accounts, Bluetooth 2.1, Microsoft Exchange, Searching SMS, MMS messages, more screen sizes
- Version 2.2 ([Froyo](#), May 2009) API 8
 - Speed and memory enhancements, new JavaScript V8 engine in Chrome, USB tethering and Wi-Fi hotspot
 - *This is lowest Android version that still actively used.*
- Version 2.3 (Gingerbread, December 2010) API 10
 - Concurrent garbage collector, video and audio improvements, near field communication (NFC), clipboard enhancements
 - *The last version only targeted for phone devices*
- Version 3.0 (Honeycomb, February 2011) API 13
 - *Optimized for tablets*, Fragments API, action bar, etc.

The History of Android (2)

- Version 4.0 ([Ice Cream Sandwich](#), October 2011) API 15
 - *Unified the tablet and smartphone SDKs*, plus improved video and a customizable launcher
- Version 4.1 (Jelly Bean, February 2012) API 16
 - Performance improvement known as “Project butter”
- Version 4.2 (Jelly Bean, November 2012) [API 17](#)
 - Camera improvements, multi-user support on tablets, unified interface layout engine, and others.
- Version 4.3 (Jelly Bean, July 2013) API 18
 - Bluetooth low energy, improved gaming graphics, etc.

The History of Android (3)

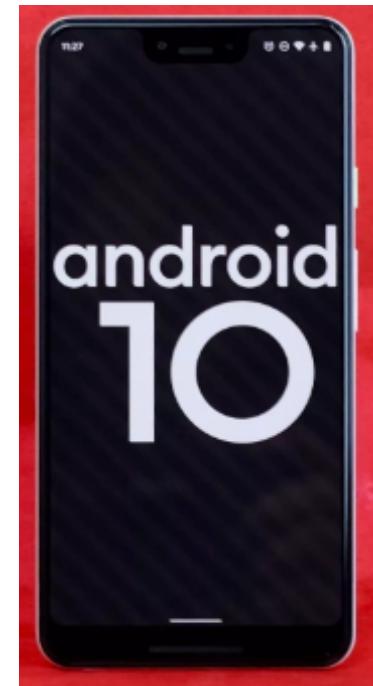
- Version 4.4 (KitKat, October 2013) API 19
 - Memory and power management, NFC, printing, storage access, SMS, rich media, accessibility
- Version 5.0 (Lollipop, November 2014) API 21-22
 - Material design, redesigned notifications, redesigned “recent apps” interface
 - *ART virtual machine replaces Dalvik virtual machine*
 - *ART translate the Dex code to machine code still app installation for improving the runtime efficiency.*

The History of Android (4)

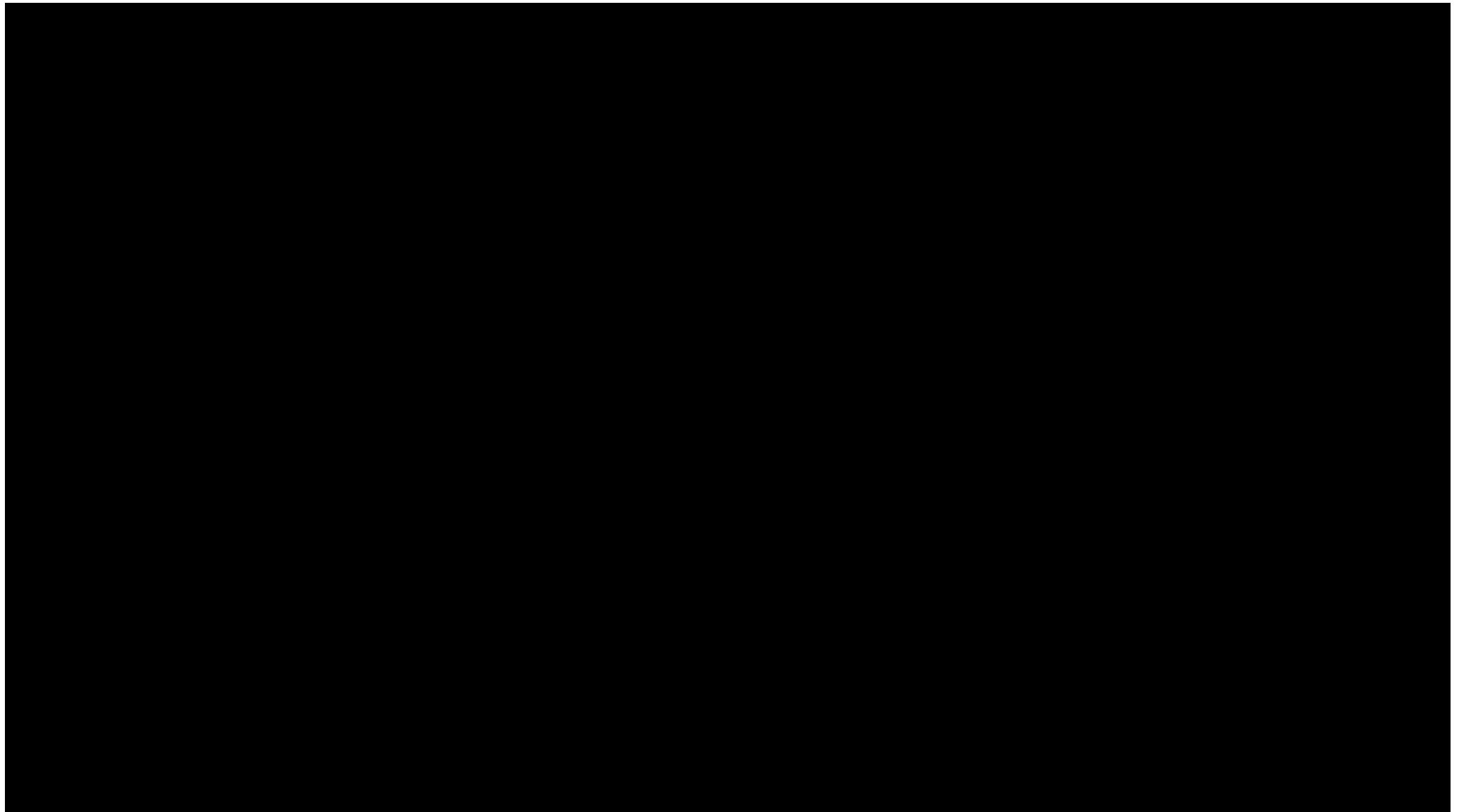
- **Version 6.0** (Marshmallow, August 2015) API 23
- **Version 7.1.1** (Nougat, August 2016) API 24-25
 - Split-screen display, Quick switching between apps, Simplified notifications, Improved battery management, etc.
- **Version 8.1** (Oreo, Sep 2017) API 25-26
 - It contains a number of major features, including notification grouping, picture-in-picture support for video, performance improvements and battery usage optimization, and support for autofillers, Bluetooth 5, system-level integration with VoIP apps, wide color gamuts, and Wi-Fi Aware.

The Latest Version of Android

- **Android 10 (Q) API 29 (Sep 3, 2019)**
 - Improved full-screen gesture system
 - Swipe up = Home
 - Swipe left or swipe right = Back
 - Support for foldable devices
 - Improvements in sharing experience
 - Support for bubbles
 - Support for Dark theme
 - Payback capture: capture audio from other apps
 - ... etc.
 - https://en.wikipedia.org/wiki/Android_10



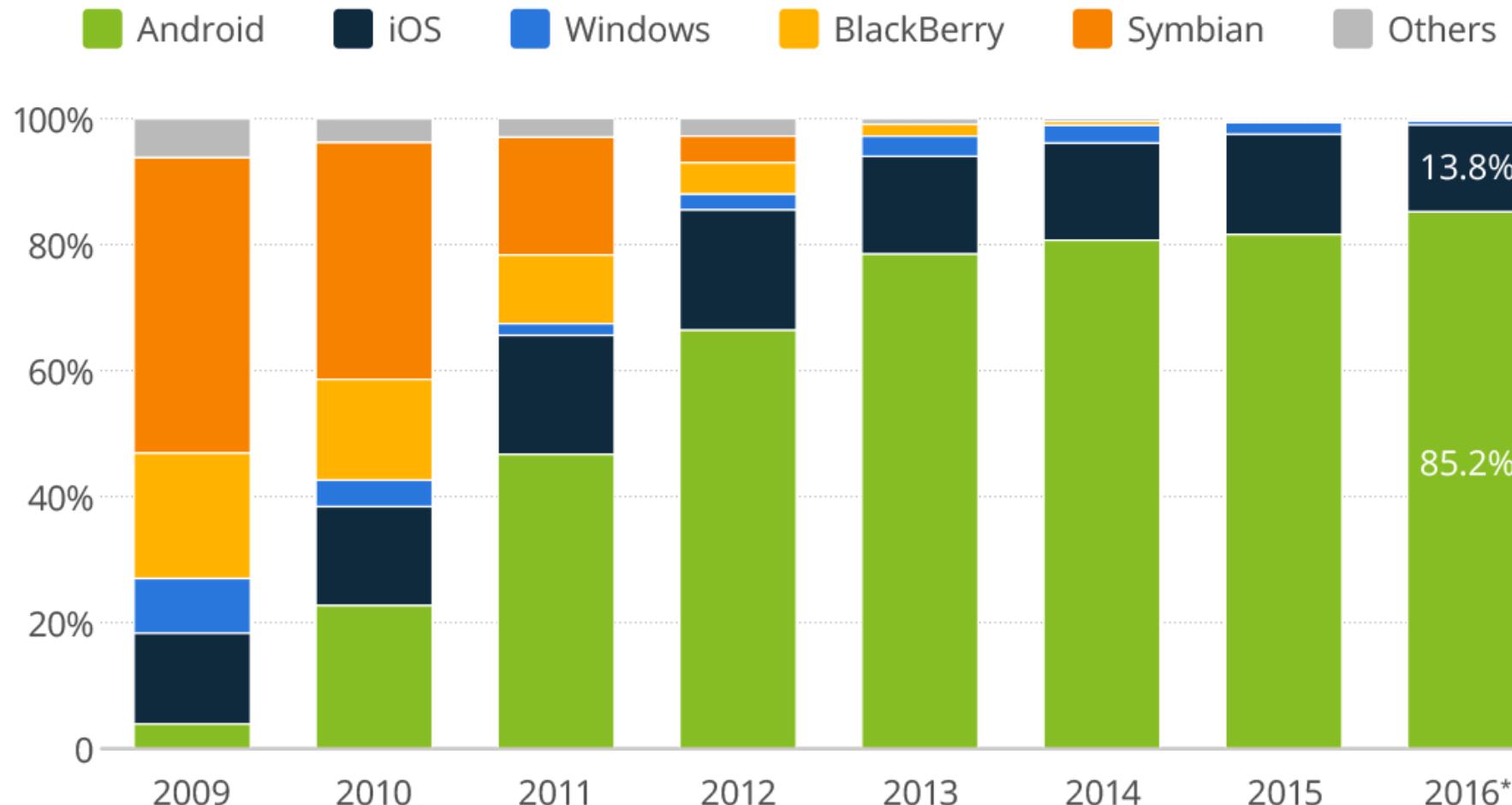
Android 10 - Full overview!



<https://www.youtube.com/watch?v=LJdX5X3FJxU>

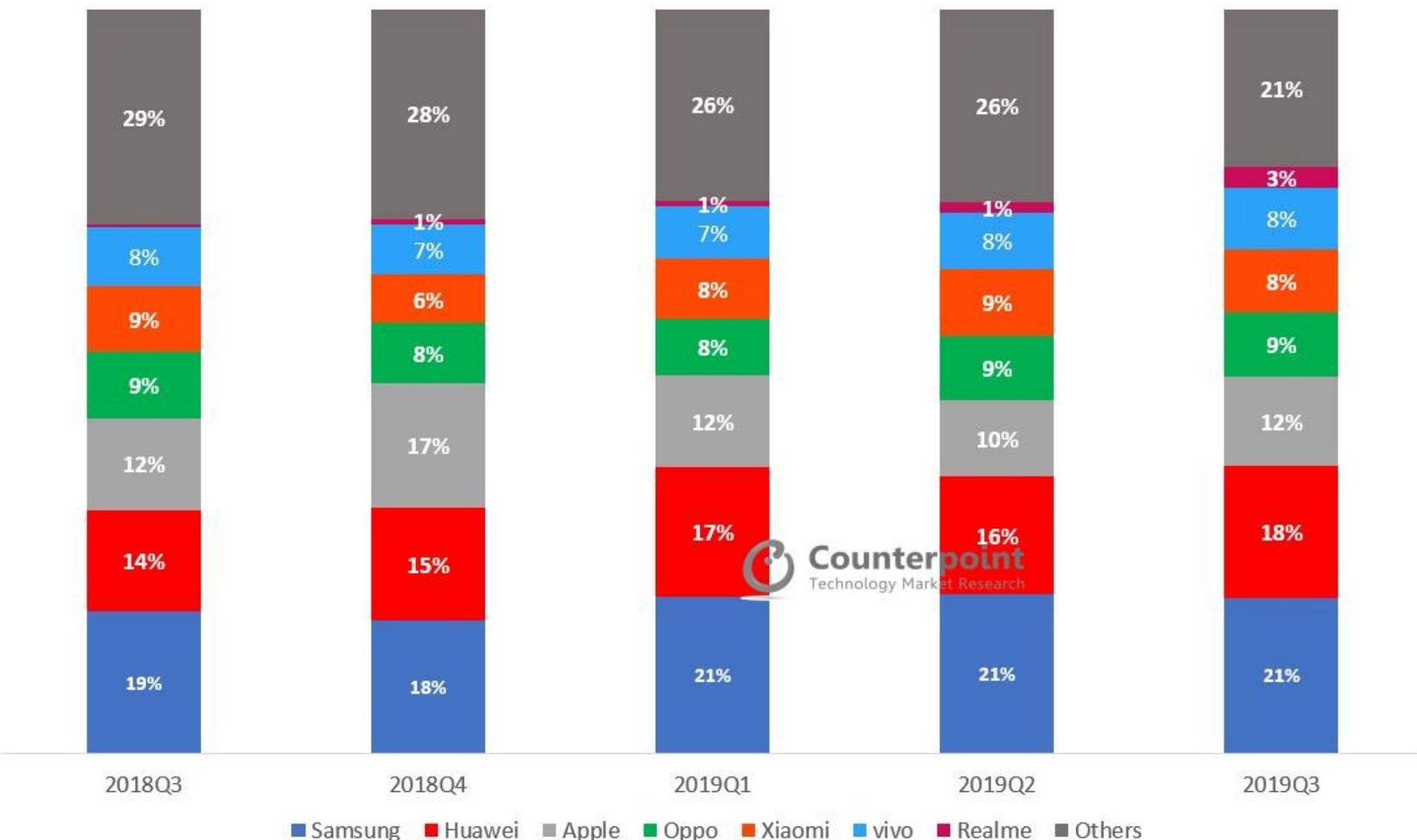
Smartphone Platform Market Share

Market share based on worldwide smartphone sales to end users



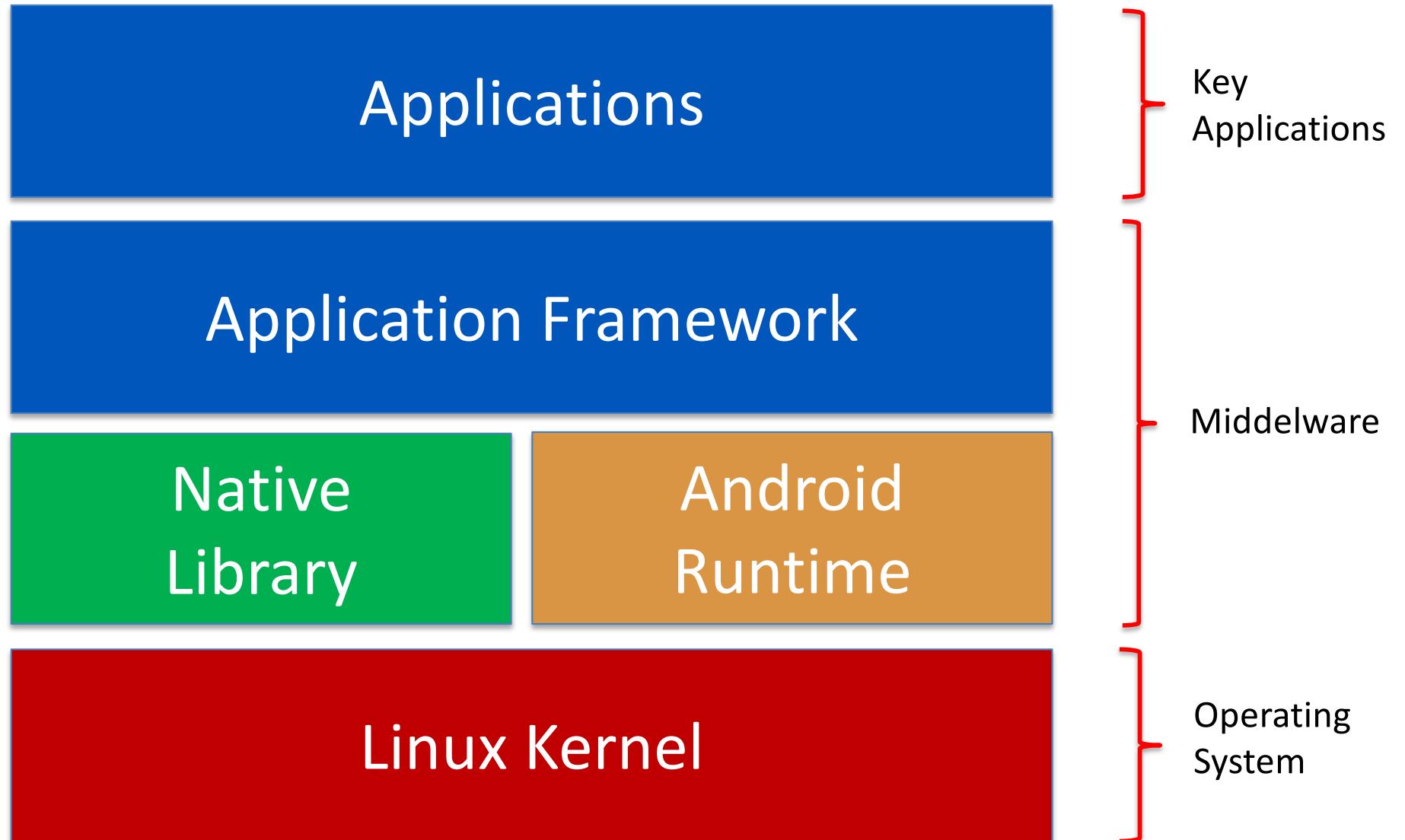
* January - June

2018 Q3 - 2019 Q3

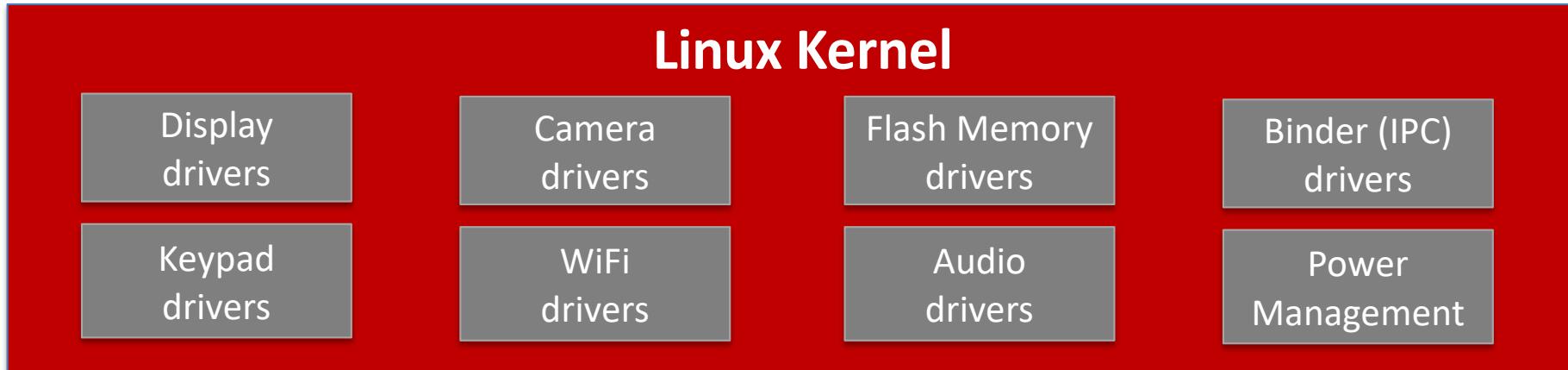


Android Software Stack

Android Software Stack

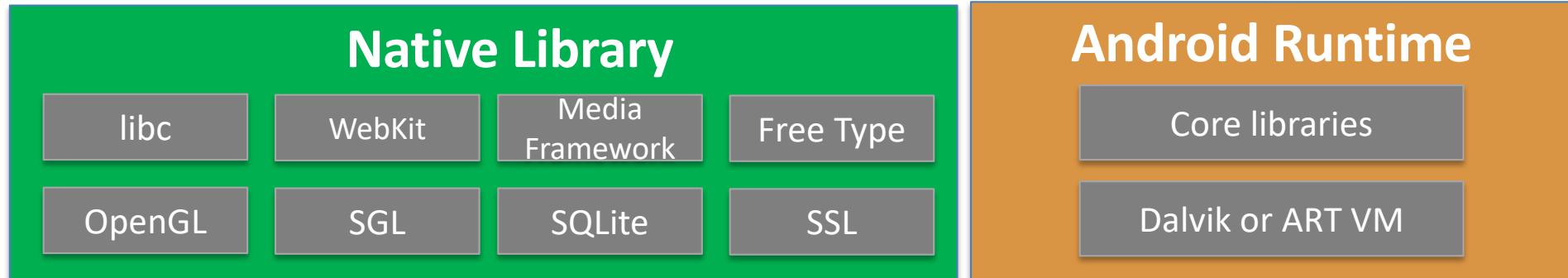


Linux Kernel



- **Standard Services**
 - Device driver
 - Security
 - Memory and Process Management
 - File & Network I/O
- **Android-specific**
 - Power Management
 - Android shared memory
 - Low memory killer
 - Interprocess communication

Native Library and Runtime Layout



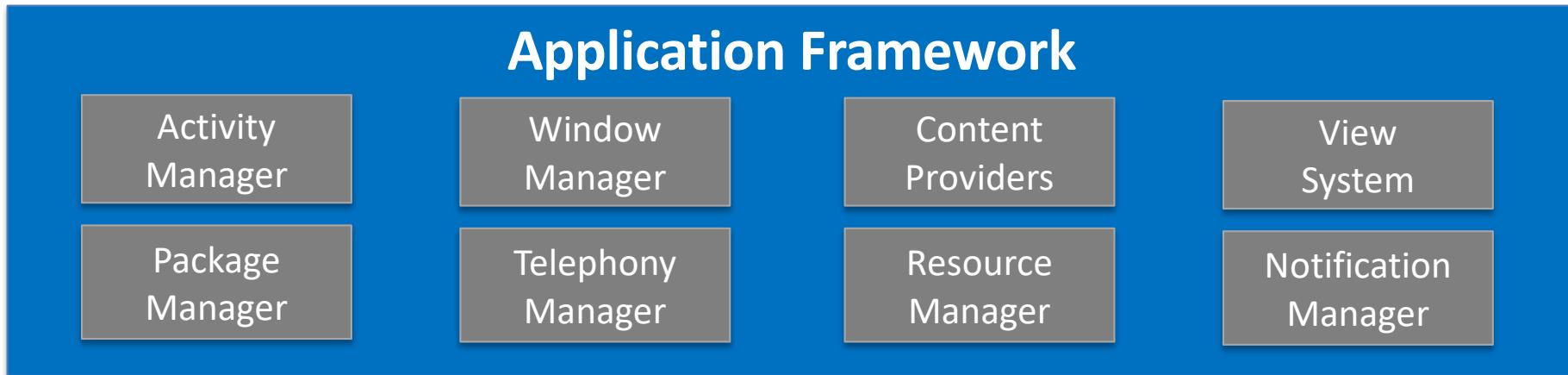
Native Library

- System C Library: Bionic
 - A super fast libc library optimized for Android
- Surface Manager
- Media Framework: Audio and Video
- WebKit library for fast HTML rendering
- OpenGL for graphics
- SQLite: Relational database engine

Android Runtime

- Dalvik or ART Virtual Machine
 - Apps are executed by the Dalvik or ART VM (Android 5.0 or after)
 - Core Java Libraries
 - Basic java classes – java.* , javax.*
 - App lifecycle – android.*
 - Internet/Web services – org.*
 - Unit testing – junit.*

Application Framework



Application Framework provides infrastructures for app developer with the help of ready-made libraries

- **Package Manager**
 - Keeps track of app packages on the devices
- **Window Manager**
 - Manages the windows comprising an app
- **View System**
 - Provides common UI elements such as icons, text entry boxes, buttons and more.
- **Resource Manager**
 - Manages non-compiled resources such as strings, graphics, and layout files
- **Activity Manager**
 - Manages app lifecycle and navigation stack
- **Content Providers: Inter-Application Data Sharing**

Applications

Applications

Home

Contacts

Phone

Browser

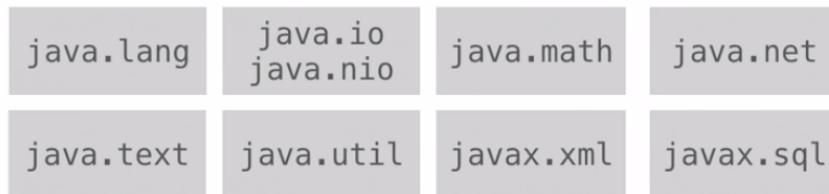
...

A set of core applications is shipped with Android

- Standard apps include
 - Home : Main Screen
 - Contacts : Contacts Database
 - Phone : Dial Phone Numbers
 - Browser : View web pages
 - Email Reader : Compose & Read email messages
- Nothing special about these apps
- You can substitute your own or 3rd party app for any of them

Program for Android with Java (1)

- The Android SDK is built around open-source Java
 - All Java 5 and 6 syntax supported
 - Most Java 7 syntax
 - Java 8 features when configured
 - Lambda expressions, method references, etc.
- Android Standard Java APIs
 - The Android SDK implements most standard Java packages



Program for Android with Java (2)

- Android Custom APIs
 - Android-specific features are in custom packages

android.app

android.widget

android.database

android.graphics

android.net

android.hardware

- API Reference
 - <https://developer.android.com/reference/>

Each Project Has a Primary Language

- Choose Java or Kotlin when creating project
- Java projects can contain Kotlin classes
- Kotlin projects can contain Java classes
- Use SDK Classes in Kotlin

In Java:

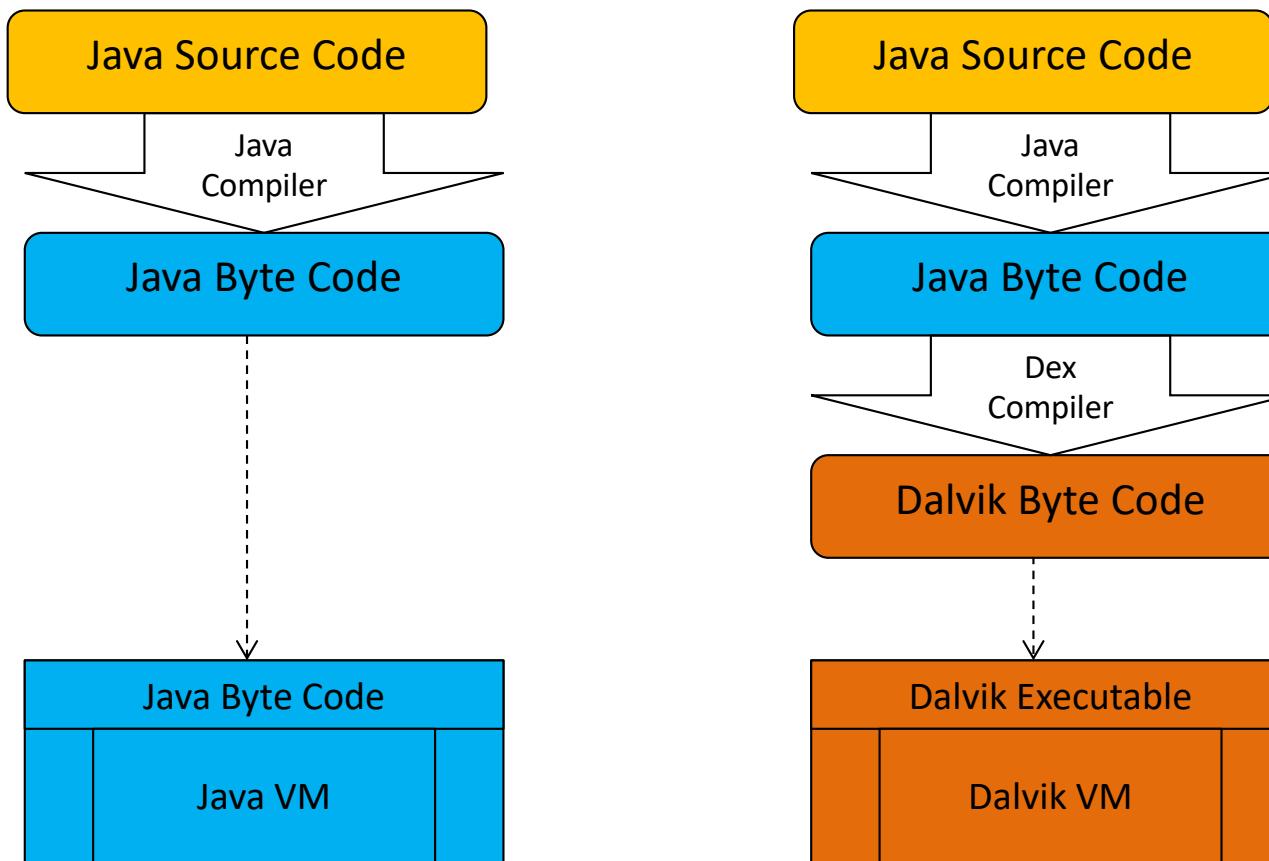
```
Button myButton = new Button(this);
```

In Kotlin:

```
val myButton = Button(this)
```

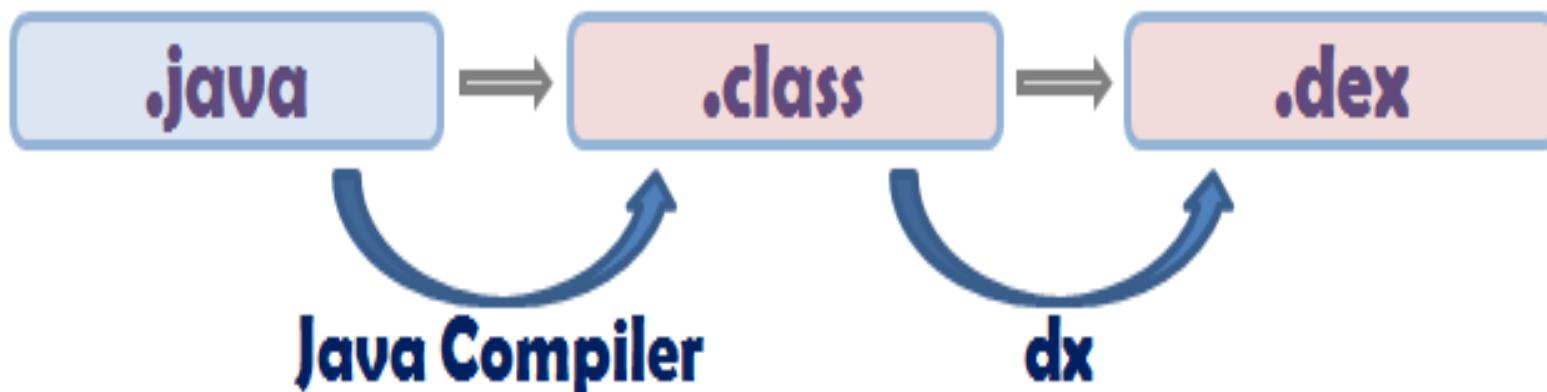
Android and Java

- Java = C++ --
- Android Java = Java SE – AWT/Swing + Android API

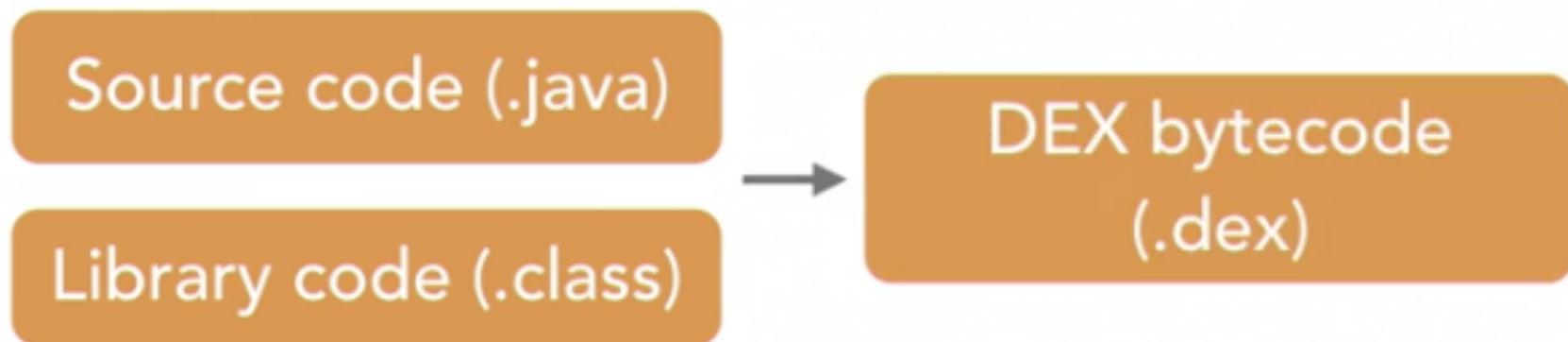


Dex Format

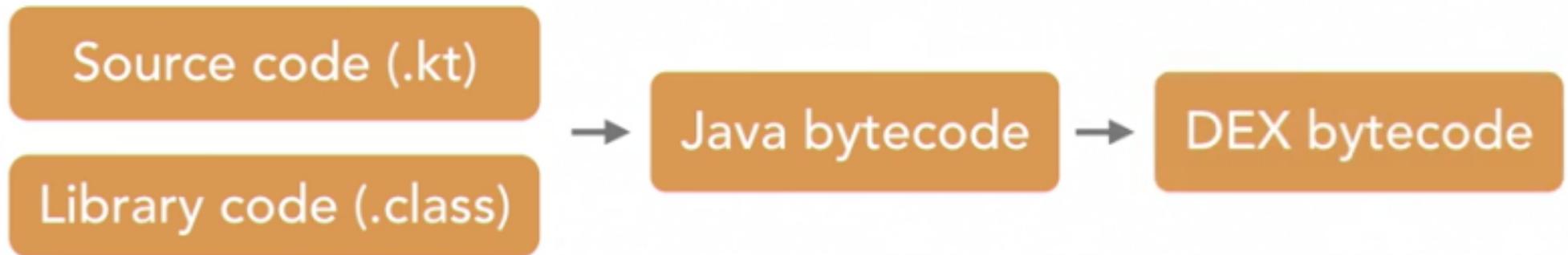
- Executing the Dalvik Executable (.dex) format
 - .dex format is optimized for minimal memory footprint.
 - Compilation



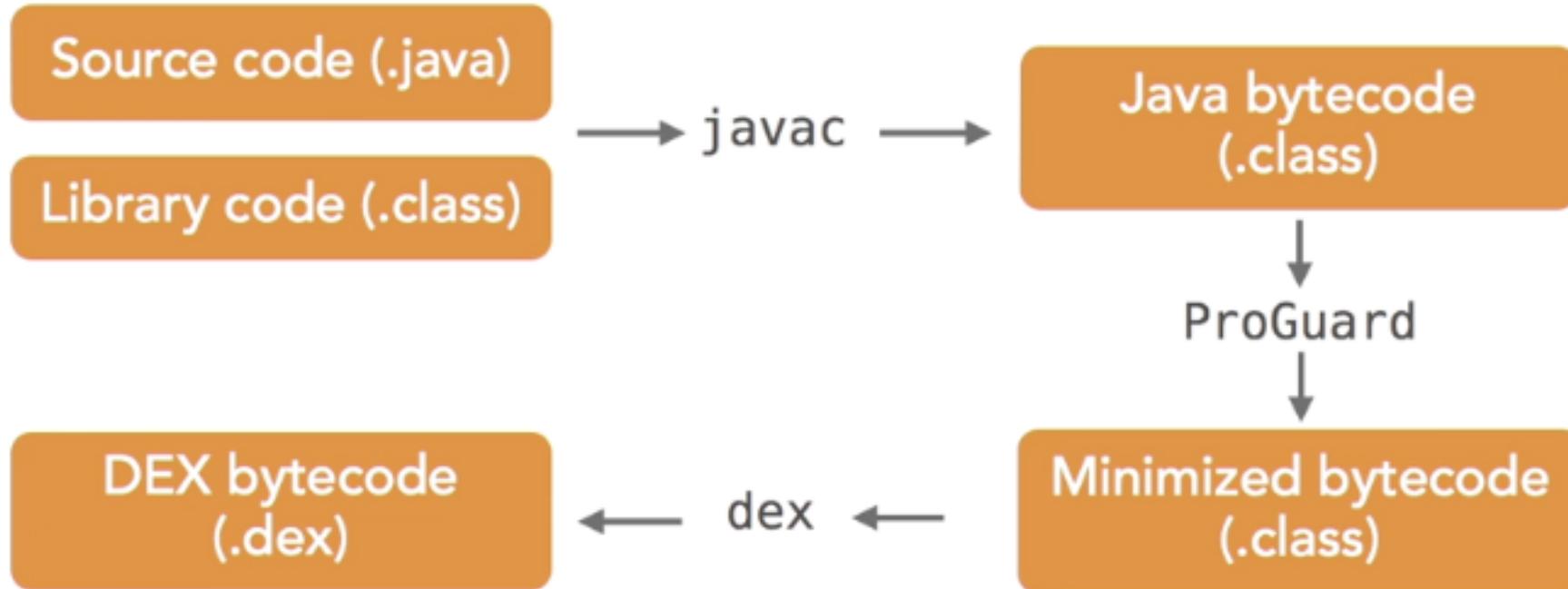
Android Build Process: Java



Android Build Process: Kotlin



Android Build Process: Legacy



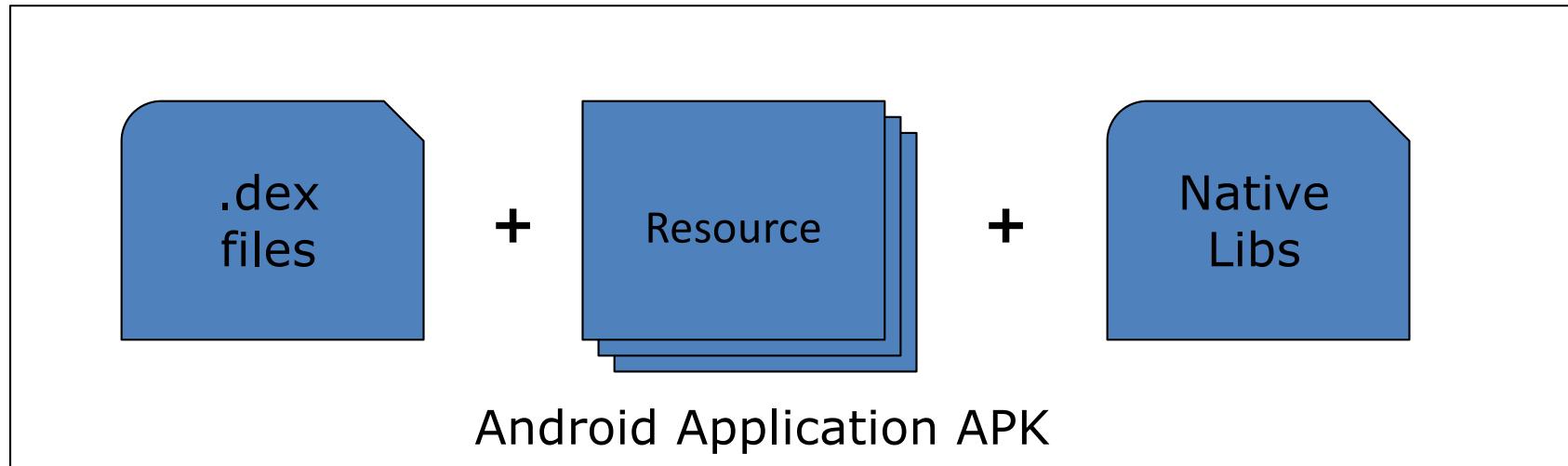
The ProGuard tool minimizes and obfuscates the bytecode so that it's harder to decompile and it's smaller.

From DEX to Machine Code

- DEX bytecode is translated to machine code on device
- Dalvik VM (Android 4.4 and prior)
 - Translation is “just-in-time” (JIT)
- ART VM (Android 5 and after)
 - Translation is “ahead-of-time” (AOT)
 - In Android 5 and 6, the translation is performed during the installation of the app on the device
 - In Android 7 (Nougat), each application will be compiled ahead of time when the first time it’s started up.

Android Application APK

- APK = .dex files + Resources + Native Libs



- The APK file must be signed
- Many markets with different policies

Android Markets

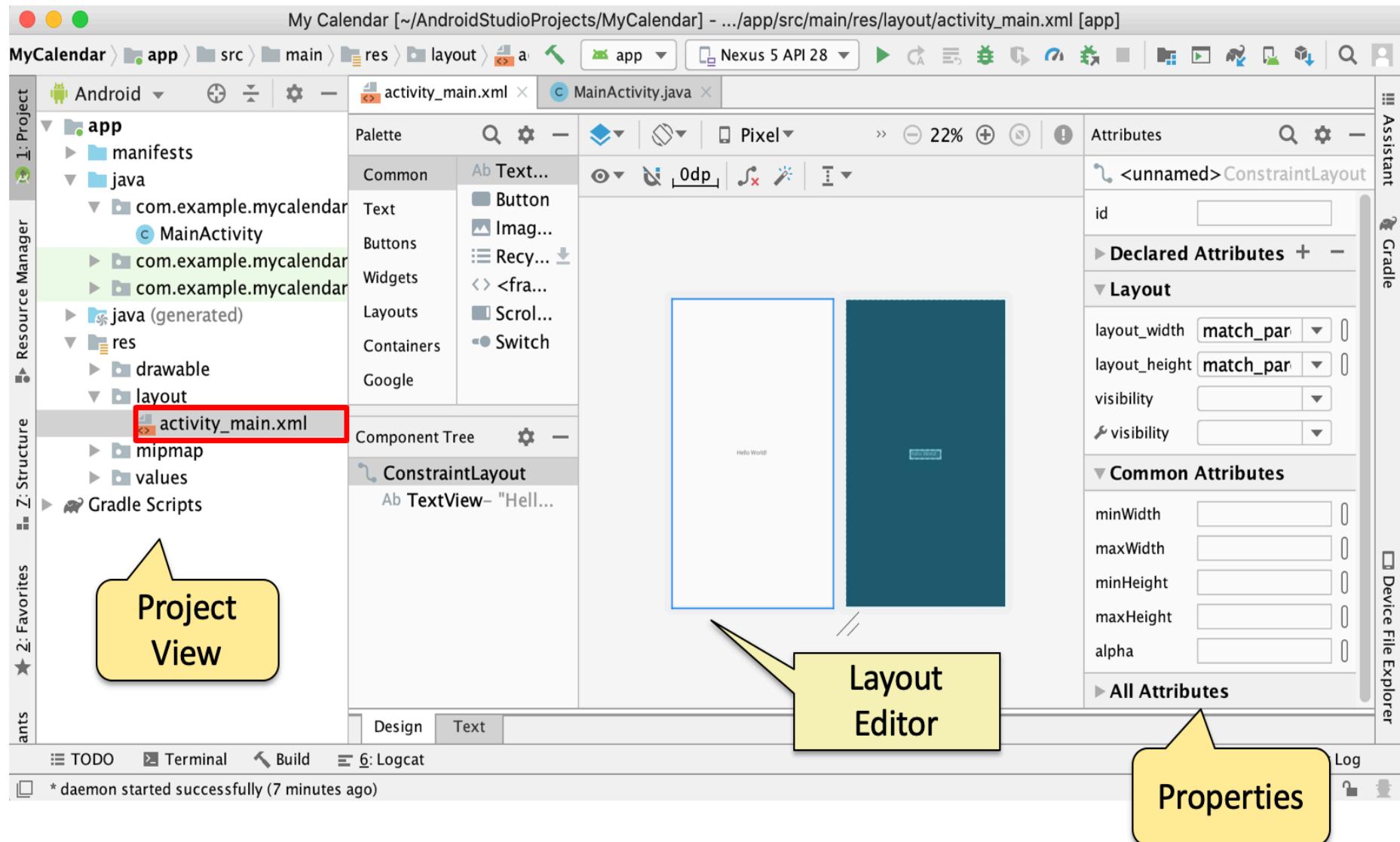
- Google Play
- Amazon Android Market
- apk.hiapk.com (安卓市场)
- Other emerging markets



Android Studio

- Starting from 10th December 2014, the official Integrated Development Environment (IDE) for Android app development is Android Studio.
- Android Studio is built on IntelliJ IDEA Community Edition from Jet Brain, which is one of the popular Java IDEs.
- At the core of Android Studio is an intelligent code editor capable of advanced code completion, refactoring, and code analysis. The powerful code editor helps us be a more productive Android app developer.
- Android Studio comes pre-configured with an optimized emulator image. Thus, we can test our apps by running them on an Android Virtual Device (AVD).

Android Studio



Course Administration

EE5415

Mobile Applications Design and Implementation

Main Feature of EE5415

- Students will form a small team with **5 members** to build their own Android Apps for smartphones.
- Lab exercises are provided to pickup the fundamental concepts and skills for design and development of Android Apps.

Prerequisites

- This course assumes you are comfortable working in **Java**.
- In addition we assume that you have some basic familiarity with **XML** (or **HTML**), **GitHub**, **networking** and **threading**.

Course Website

- Course Website:
 - <http://www.ee.cityu.edu.hk/~lmpo/ee5415/index.html>
 - Students should check the course website regularly for announcements about the course.
 - Copies of the lecture, assignment, project information are all available in course website with various formats.
- Messages, Schedule, Projects, Links
 - Username: **students**
 - Password: **android10**

Grading

- 5% - Labs with short question assignments
- 5% - Individual Project in Lab05 (Week 9)
- 5% - Test (Week 6)
- 30% - Mid-Term Exam with programming (Week 10)
- 15% - Group Project (Weeks 5, 12, 13)
- 40% - Final Exam
- To pass the course, students are required to achieve at least 30% in course work and 30% in the Final examination.
- Also, 75% laboratory attendance rate must be obtained.

Lab Attendance Policy

- To pass this course, a laboratory attendance of at least 75% is required.
- As there are about 10 lab sessions, **absence of 3 or more lab sessions** is considered as not satisfying the attendance requirements.

Electronic Submission

- Hardcopies are not required for some of the assignments.
- Softcopies such as MS-Word file and associated files are required to send to the course email account before 6:00PM of the due date.
- EE5415 course email address:
 - ee5415@gmail.com
- In order to let us more easy to handle your mails, please use the following format for your mail's subject:
 - EE5415 Assignment Name - Student Name (Student Number)
 - For examples:
 - EE5415 Project Proposal – Group 1 Compass
 - EE5415 Lab1 Assignment - Chan Chi Ming (501234565)

Late Submission

- All assignments and report mush be received by the course email account before 11:00PM on the due date.
- NO late homework is accepted without previous arrangement with the instructor.
- If approved, late HW receives 10% per business day penalty.
- Use the assignment name, your name and student number as subject for sending your work to ee5415@gmail.com account by email.
- Students may work together on the homework, but copying is unacceptable.
- **Work to learn. Don't work for marks.**

Cheating

- In particular copying your fellow classmate's assignments or programs, is a very serious offense! If you are found cheating, you will automatically get an F grade in this course and your act will be reported to the Department for necessary disciplinary actions.
- Please don't let others copy your assignments or programs as we don't have a way to tell who is copying who and you may be liable to the penalties.

Textbook

- No textbook for this course
- Most of the reference materials of this course can be found in Android Developer Website:
 - <http://developer.android.com>
- Essential Links are also available in the Links webpage of the course website.

Readings

- Readings will be assigned for each class. They will be designated on the course website as well as in the preceding lecture.
- PLEASE read over the indicated sections before class. Be prepared to discuss the subject intelligently and/or ask questions about material you don't understand.

Course Project

- Students will work in teams with **five students per group** to implement and deploy Android Mobile apps of their proposed applications.
- Each team is expected to apply best software development practices such as using software configuration management, and select appropriate supporting tools.

Assessment of the Course Projects

- Project Proposal and Presentation on week 5.
- Final Presentation with demo and Final Report on week 12 or 13
- More details can be found in the project webpage of the course website.

Proposal and Presentation (1)

- The proposed apps must be non-trivial, ideally incorporating multiple significant features of the Android platform. The proposed app should satisfy at least one or two of the following categories:
 - humanitarian (i.e. it improves quality of life in some significant way)
 - monetizable (i.e. it could generate a profit)
- The proposal must include research to similar apps that are available in the Android Markets. Students are required to justify that their idea either adds a significant feature, or would be implemented in a way that is a significant improvement over the existing app.
- In the proposal, express the vision and scope of the proposed app.
- Competitive Analysis (who are competitors, how successful are they, how is your idea different)

Proposal and Presentation (2)

- Template of the project proposal: App Name
 - Names of students in the team
 - Summary of the project (about half page)
 - Preliminary list of functional requirements.
 - Explain the major functions of the proposed app and the related theory.
 - Software architecture and User Interface Design (Some draft screen layouts to get you thinking about layout. The layouts can be handwritten for this milestone.)
 - Market Research with competitive Analysis (who are competitors, how successful are they, how is your idea different)
- Project Teams are required to present their proposed Android Mobile Apps with **5 minutes** on week 4.
- [The project proposal template can be downloaded here.](#)
- **Deadline:** The proposal must be sent to the ee5415@gmail.com before 11:00pm on 19 Feb 2020.

Final Presentation and Report

- Implement a complete version of your app. This version should be feature complete and fully polished.
- Project Teams are required to give an oral presentation of their completed apps on weeks 12 or 13.
- The presentation is about **15 minutes** with 2 minutes for Q&A.
- **Students are required to present complete story of the developed app including the background, app's features, performance evaluation, demo and conclusion.**
- A short video to introduce the developed app is required which can demonstrate the main features of the developed app.
- The requirements are same as the progress presentation and Progress Report as listed above.
- [The project final report template can be downloaded here.](#)
- A final report with source code, ppt, mini-poster and promotion video of the app are required to submit on week 14 with all details of the project.

Project Grouping

- Students are required to form the project groups with **5 students per group** on or before the Friday of week 2.
- Submit your group member list to the course gmail account at ee5415@gmail.com before 11.00pm on next Friday.

Lab Session

- Locations: Yeung Building, Room P1406 and Room P1442
 - A list of lab-assigned students can be found on the course website.
- Lab01 : Android Development System
 - Before students go to the lab to practice, we will first introduce the lab.