

# INTRODUCTION TO ANDROID

PREPARED BY:

SUSAN S. CALUYA, MSC



# Course Outcomes

**At the end of the lesson students should be able to:**

- ▶ Explain the android ecosystem
- ▶ Analyze android platform architecture
- ▶ Identify application development challenges

# Topics



Android is an ecosystem



Android platform architecture



Android version



Application development



# **Android as an Ecosystem**

# What is Android?

- ↳ Mobile operating system based on Linux kernel
- ↳ User Interface for touch screens
- ↳ Used on over 80% of all smartphones
- ↳ Powers devices such as watches, TVs, and cars
- ↳ Over 2 Million Android apps in Google Play store
- ↳ Highly customizable for devices / by vendors
- ↳ Open source

# Android User Interaction

- ↳ Touch gestures: swiping, tapping, pinching
- ↳ Virtual keyboard for characters, numbers, and emoji
- ↳ Support for Bluetooth, USB controllers and peripherals

# Android and Sensors

Sensors can discover user action and respond

- ↳ Device contents rotate as needed
- ↳ Walking adjusts position on map
- ↳ Tilting steers a virtual car or controls a physical toy
- ↳ Moving too fast disables game interactions

# Android Home Screen

- ↳ Launcher icons for apps
- ↳ Self-updating widgets for live content
- ↳ Can be multiple pages
- ↳ Folders to organize apps
- ↳ "OK Google"



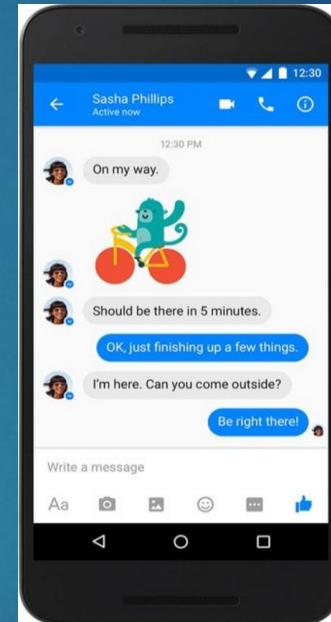
# Android App Examples



Pandora



Pokemon GO

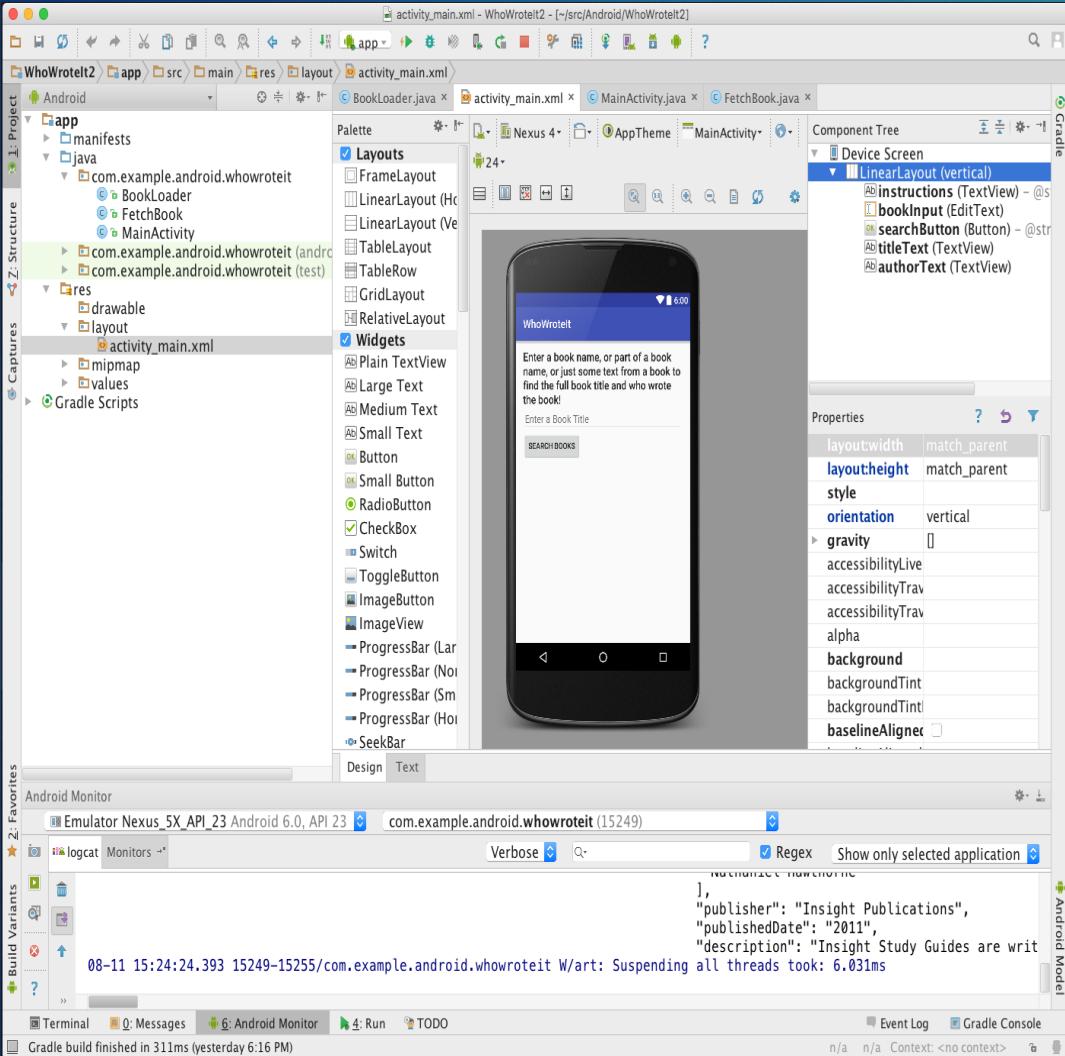


Facebook  
Messenger

# Android Software Development Kit (SDK)

- ↳ Development tools (debugger, monitors, editors)
- ↳ Libraries (maps, wearables)
- ↳ Virtual devices (emulators)
- ↳ Documentation ([developers.android.com](http://developers.android.com))
- ↳ Sample code

# Android Studio



- Official Android IDE
- Develop, run, debug, test, and package apps
- Monitors and performance tools
- Virtual devices
- Project views
- Visual layout editor

# Google Play Store

Publish apps through Google Play store:

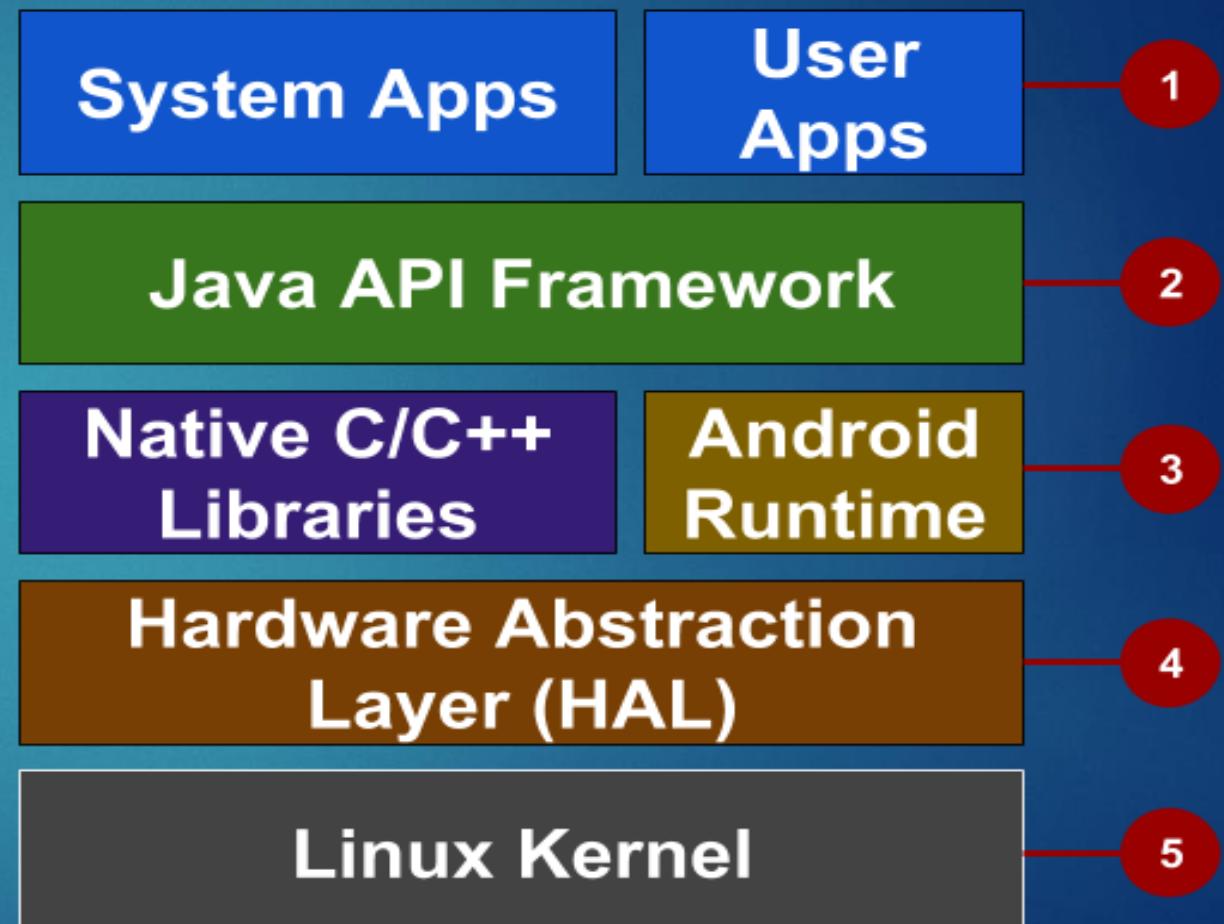
- ↳ Official app store for Android
- ↳ Digital distribution service operated by Google



# **Android Platform Architecture**

# Android Stack

1. System and user apps
2. Android OS API in Java framework
3. Expose native APIs; run apps
4. Expose device hardware capabilities
5. Linux Kernel



# System and User Applications

- ↳ System apps have no special status
- ↳ System apps provide key capabilities to app developers

Example:

Your app can use a system app to deliver a SMS message.

# Java API Framework

The entire feature-set of the Android OS is available to you through APIs written in the Java language.

- ↳ View class hierarchy to create UI screens
- ↳ Notification manager
- ↳ Activity manager for life cycles and navigation

# Android Runtime

Each app runs in its own process with its own instance of the Android Runtime.

# C/C++ Libraries

- Core C/C++ Libraries give access to core native Android system components and services.

# Hardware Abstraction Layer (HAL)

- ↳ Standard interfaces that expose device hardware capabilities as libraries

Examples: Camera, bluetooth module

# Linux Kernel

- ↳ Threading and low-level memory management
- ↳ Security features
- ↳ Drivers

## **Android Versions**

# Older Android Versions

Codename	Version	Released	API Level
<b>Honeycomb</b>	3.0 - 3.2.6	Feb 2011	11 - 13
<b>Ice Cream Sandwich</b>	4.0 - 4.0.4	Oct 2011	14 - 15
<b>Jelly Bean</b>	4.1 - 4.3.1	July 2012	16 - 18
<b>KitKat</b>	4.4 - 4.4.4	Oct 2013	19 - 20
<b>Lollipop</b>	5.0 - 5.1.1	Nov 2014	21 - 22

[Android History and Platform Versions](#)

for more and earlier versions before 2011

# Newer Android Versions

Codename	Version	Released	API Level
<b>Marshmallow</b>	6.0 - 6.0.1	Oct 2015	23
<b>Nougat</b>	7.0 - 7.1	Sept 2016	24 - 25
<b>Oreo</b>	8.0 - 8.1	Sept 2017	26 - 27
<b>Pie</b>	9.0	Aug 2018	28

# **Application Development**

# What is Android Application?

- ↳ One or more interactive screens
- ↳ Written using Java Programming Language and XML
- ↳ Uses the Android Software Development Kit (SDK)
- ↳ Uses Android libraries and Android Application Framework
- ↳ Executed by Android Runtime Virtual machine (ART)

# Challenges of Android Development

- ↳ Multiple screen sizes and resolutions
- ↳ Performance: make your apps responsive and smooth
- ↳ Security: keep source code and user data safe
- ↳ Compatibility: run well on older platform versions
- ↳ Marketing: understand the market and your users  
(Hint: It doesn't have to be expensive, but it can be.)

# Application Building Blocks

- ↳ Resources: layouts, images, strings, colors as XML and media files
- ↳ Components: activities, services, and helper classes as Java code
- ↳ Manifest: information about app for the runtime
- ↳ Build configuration: APK versions in Gradle config files



# Thank You!