

# MODULE 1

# INTRODUCTION

# TABLE OF CONTENT

- Android History
- Android Software Stack
- Android Studio
- Activity
- Creating an Android Application
- Android Project Structure
- Android Manifest File
- Dalvik Virtual Machine
- Application Package File

# TABLE OF CONTENT

- Gradle
- Genymotion
- LogCat
- Webography

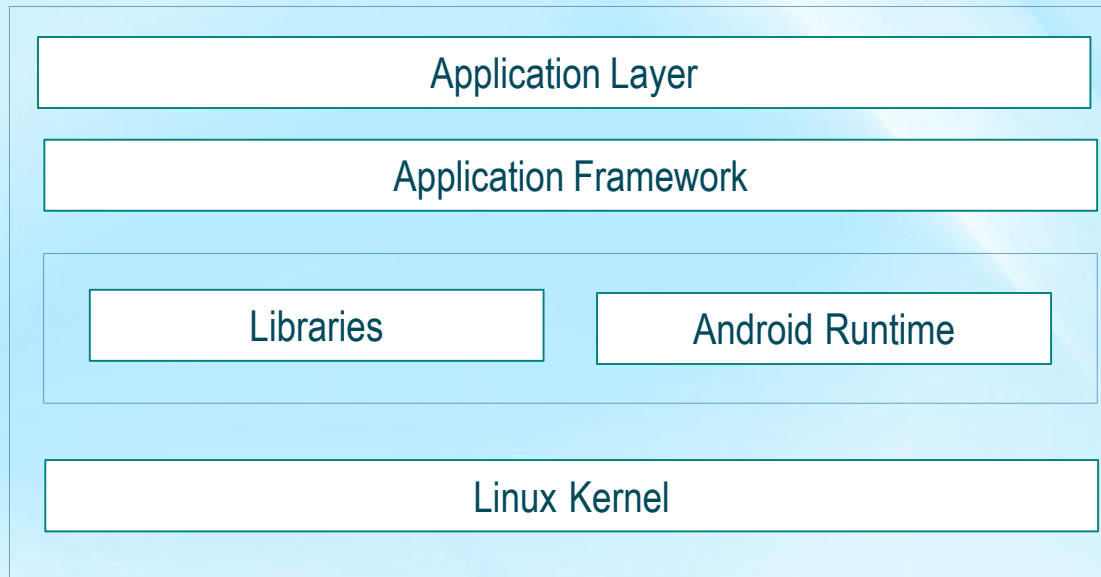
# Android History

- ▶ Google's open source and free Java-based platform for mobile development
- ▶ Using the Android software development kit (SDK)
- ▶ To publish on the Android market
- ▶ First developed by Android (Palo Alto California)
- ▶ Google bought the company in November 2007

# Android History

Platform Version	Code Name
Android 1.5	Cupcake
Android 1.6	Donut
Android 2.0 and 2.1	Eclair
Android 2.2	Froyo
Android 2.3	Gingerbread
Android 3	Honeycomb
Android 4.0	Ice Cream Sandwich
Android 4.1 ⇒ 4.3	Jelly Bean
Android 5.0	Lollipop
Android 6.0	Marshmallow
Android 7.0	Nougat

# Android Software Stack



# Android Software Stack

- ▶ Linux Kernel : Device drivers
  - Display
  - Camera
  - Keypad
  - Wi-Fi
  - Flash memory
  - Audio
- ▶ Libraries
  - WebKit library : browser support
  - FreeType library : font support
  - SQLite library : database support
  - Media library : recording and playback of audio and video formats
  - Surface Manager library : 2D and 3D graphics support
  - ...

# Android Software Stack

## ▶ Android Runtime

- Enables developers to write Android applications using java
- Contains
  - Core Android libraries
    - Providing most of the functionalities of the core Java libraries
  - Dalvik virtual machine

## ▶ Application Framework

- Provides classes to manage
  - User interface
  - Application resources
  - Abstraction for hardware access



# Android Software Stack

- ▶ Application Layer
  - Displays the application along with the built-in applications provided by the device itself

# Android Studio

- ▶ To build Android applications
  - Automatically creates the necessary Android files such as
    - Java files
    - XML resource and layout files
    - Manifest files
  - [\*http://developer.android.com/sdk/index.html\*](http://developer.android.com/sdk/index.html)

# Activity

- ▶ An application = one or more activities
- ▶ Activity
  - Usually represents a single screen
  - Consists of one or more user interface (GUI) controls
    - E.g, TextView, Button, EditText, ...
  - Enables user interaction with the application
- ▶ Separation of the presentation layer (View) from business logic
  - Separation of the user interface from the action code

# Activity

- ▶ View: Activity XML file
  - Defines the user interface of the application
  - Contains GUI controls
- ▶ Business logic: Activity Java file
  - Contains action code of the GUI controls
    - Event handling
    - Processing of data entered by the user

# Creating an Android Application

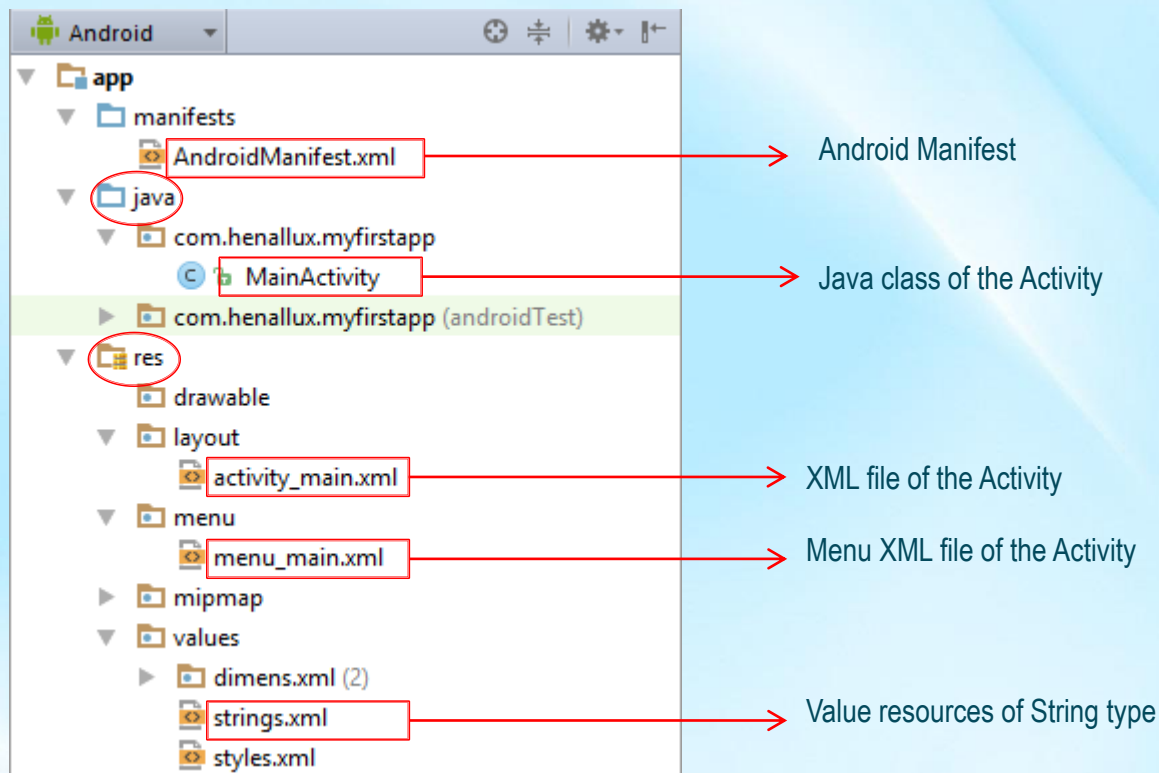
- ▶ Application name and Company domain
  - Must be unique
  - Once an application is published, cannot be changed
  - E.g,
    - Application name: MyHelloApp
    - Company domain: henallux.com
    - ⇨ Package name: *com.henallux.myHelloApp*

# Creating an Android Application

## ► Target Android Devices

- Different platforms
  - Phone and Tablet
  - Wear
    - Small powerful devices worn on the body
  - TV
  - Android Auto
    - To be operated in automobiles through the dashboard's head unit
  - Glass
    - Wearable technology with an optical head-mounted display
- Minimum SDK
  - The minimum version of the Android platform required by the application to run

# Android Project Structure





# Android Project Structure

- ▶ manifests folder
  - Contains AndroidManifest.xml
    - Central configuration file for the application
- ▶ java folder
  - Contains a directory corresponding to the package name
    - Containing Java source files of activities



# Android Project Structure

## ▶ res folder

- All application resources
  - Images
  - Layout files
  - String files
- Rather than hard coding image or string into the application, better
  - To create respective resource in the res folder
  - To include its reference in the application (ID included in the R.file, see further)
- drawable folders
  - Icons and graphics resources according to the screen resolutions
    - Drawable-xhdpi folder : for 320dpi
    - Drawable-hdpi folder : for 240dpi
    - Drawable-mdpi folder : for 160dpi
    - Drawable-ldpi folder : for 120dpi

# Android Project Structure

- ▶ res folder (continue)
  - layout folder
    - Stores the activity layout files
    - One XML file per activity
  - menu folder
    - Stores the menu layout files
    - One XML file per activity
  - values folder
    - Stores all the values resources
    - Many types : dimensions, strings, color, ...

# Android Manifest File

- ▶ Defines the overall structure and information to run the application
  - Each activity
    - The entry point: the first activity to launch
  - Services
    - Tasks in the background
  - Intents
  - Needed permissions
  - Meta data
    - Icons
    - Labels
  - Information required in building and packaging
    - For installing and deploying the application

# Android Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.henallux.myfirstapp" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme"
        android:name="MyApplication"
    >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Application ID

For each activity

# Android Manifest File

## ► Other tags

- service tag
  - For each process running in the background
- uses-permission tag
  - To declare permissions that the application needs to run
  - E.g, `<uses-permission android:name="android.permission.CAMERA" />`
    - If the application needs to use camera

# Dalvik Virtual Machine

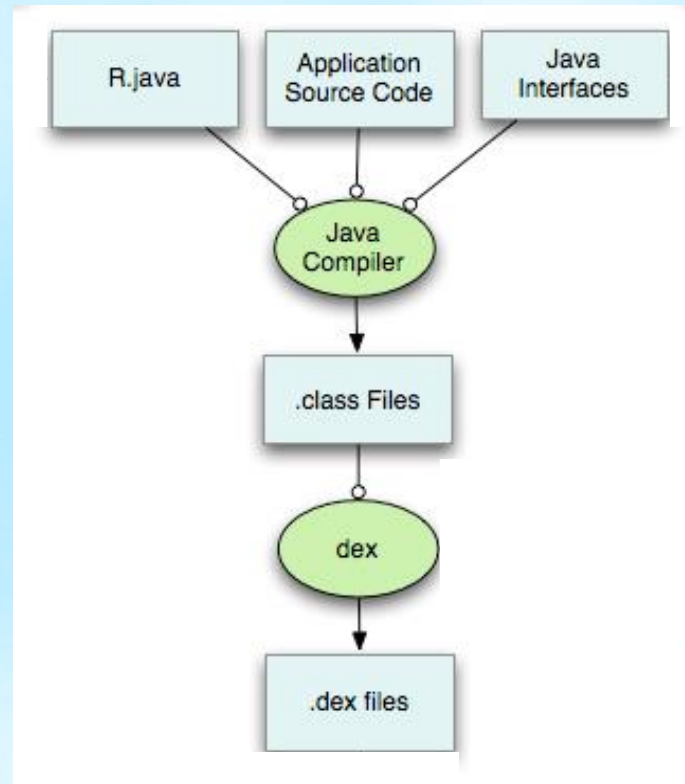
- ▶ The Android platform's virtual machine
  - Provides environment to deploy and run Android applications
- ▶ Optimized for mobile devices with limited
  - Battery
  - Memory
  - Computation capability

# Dalvik Virtual Machine

- ▶ When running an application
  1. Android SDK accesses all layout and variable information in XML files
    - ↳ converts it into Java source code placed in **R.java** file
  2. R.java file compiled into Java byte code files (.class files)
    - ↳ + the dx tool converts it into Dalvik byte code :  
Dalvik Executable format (**.dex**) optimized for
      - Efficient Storage
      - Low memory consumption



# Dalvik Virtual Machine



Source: [developer.android.com/sdk/installing/studio-build.html](http://developer.android.com/sdk/installing/studio-build.html)

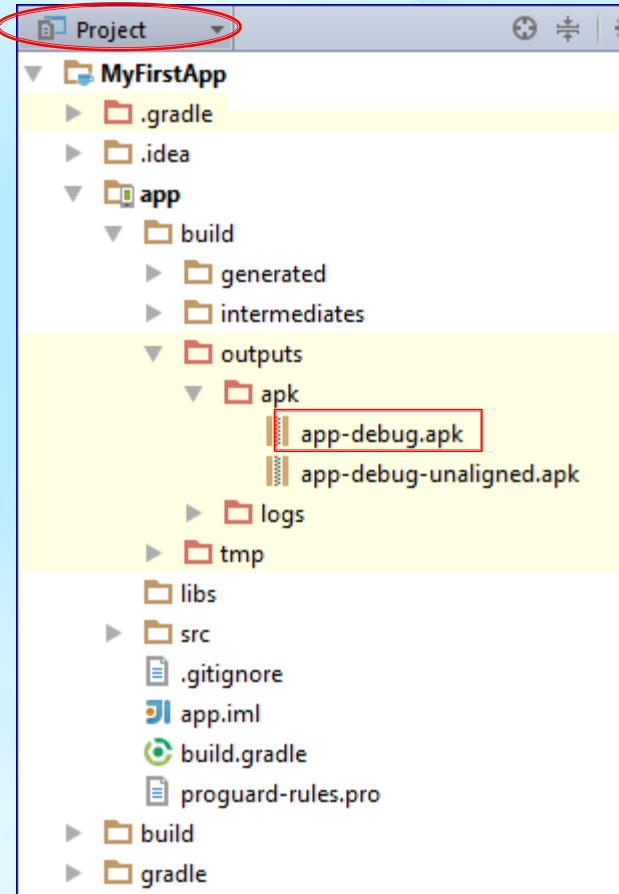


# Application Package File

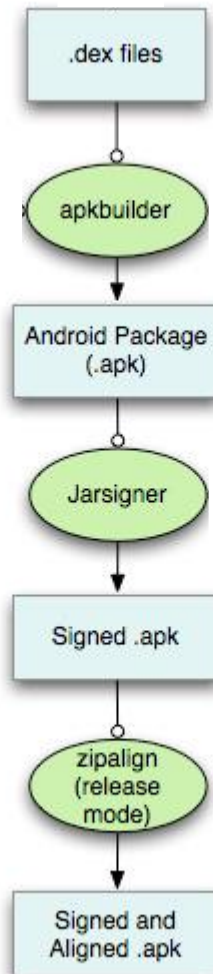
- ▶ The Android applications are not deployed in .dex format
- ▶ The dex code is bundled into an APK file
  - With required data and resources
  - Including the AndroidManifest.xml
  - apk file extension
- ▶ APK file used to distribute Android application
  - To install it on mobile device or emulator

# Application Package File

In Android studio: change to Project view



# Application Package File



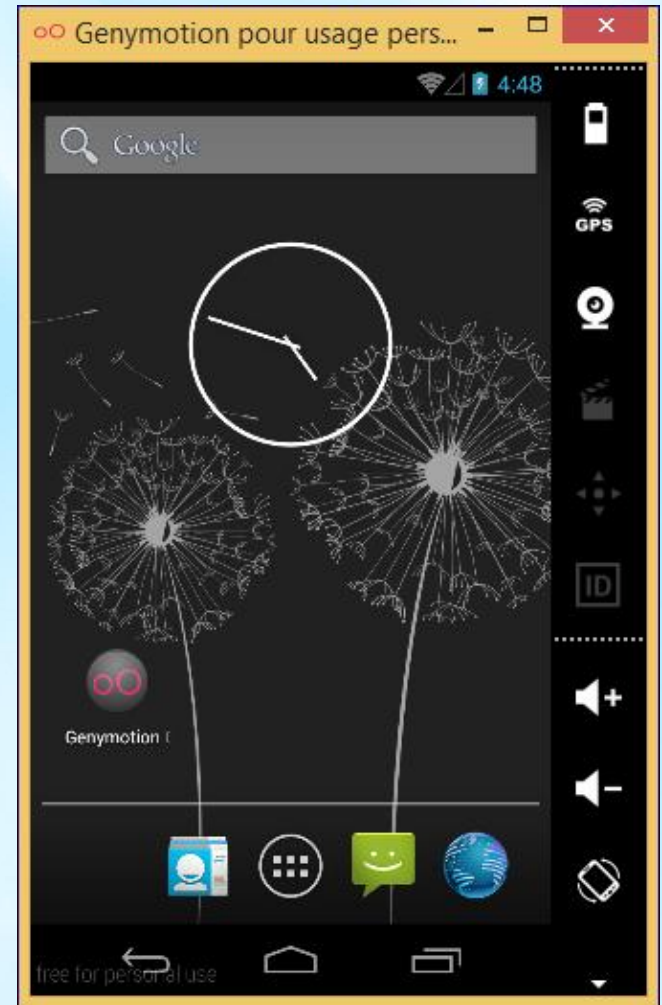
Source: [developer.android.com/sdk/installing/studio-build.html](http://developer.android.com/sdk/installing/studio-build.html)

# Gradle

- ▶ Build automation tool
  - Determines the order in which tasks can be run
- ▶ Designed for multi-project builds which can be quite large
- ▶ Incremental builds
  - Intelligently determines which parts of the build tree are up-to-date
    - ⇒ those parts will not be re-executed

# Genymotion

- ▶ Much more efficient emulator
  - To test applications with best performance
  - User-friendly interface
- ▶ [www.genymotion.com](http://www.genymotion.com)



# LogCat

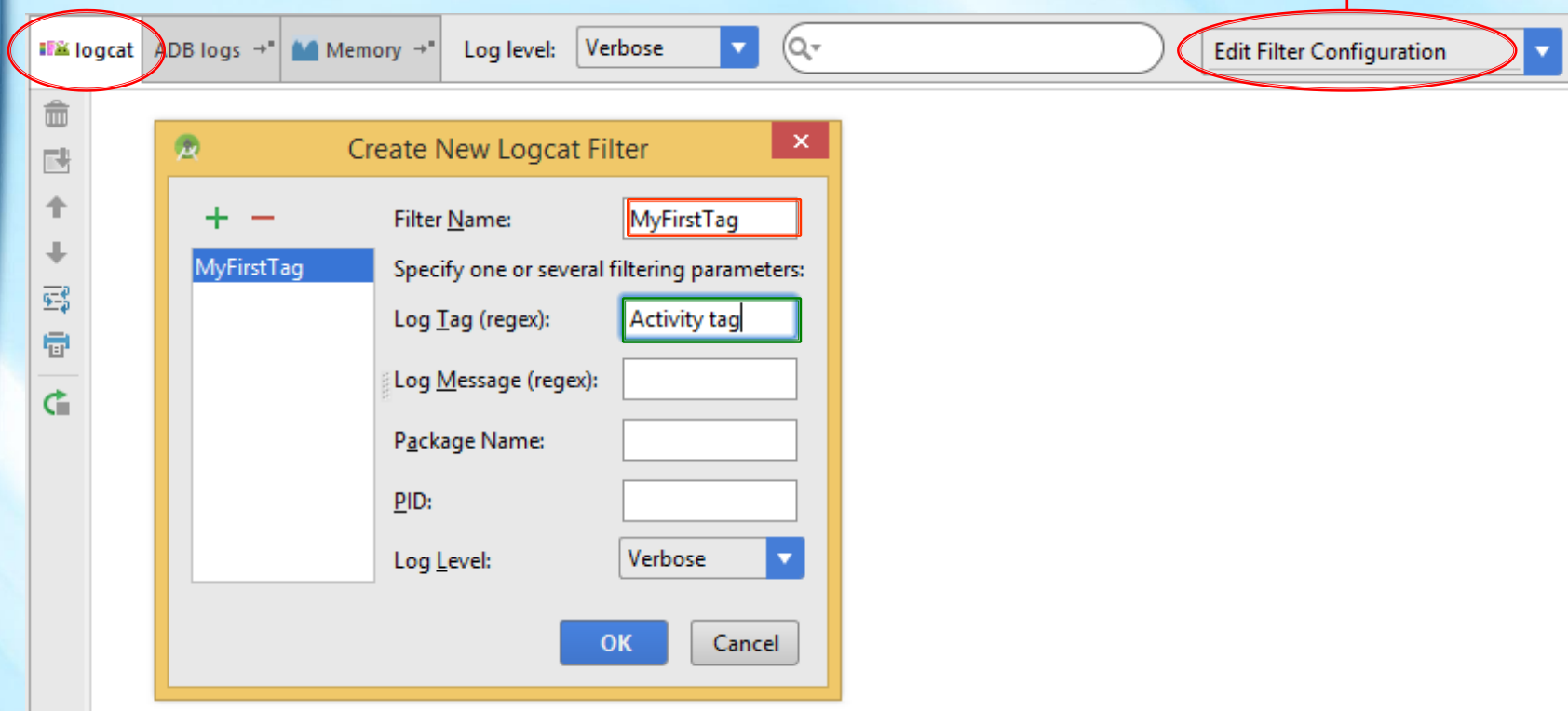
- ▶ Commonly used to debug an android application
- ▶ Through **android.util.Log** class
  - Using **Log.i** ("Tag Id", "Message to display in the LogCat window")
  - E.g,

```
Log.i("Activity tag", "Launching of main activity");
```

# LogCat

- Filter messages to display through LogCat window

To create new filter



# LogCat

Selected filter

The screenshot shows the LogCat window in Android Studio. The top bar includes tabs for 'logcat', 'ADB logs', and 'Memory'. The 'Log level' is set to 'Verbose'. A search bar is present. The 'Selected filter' dropdown is set to 'MyFirstTag'. The log entry is: '09-02 19:07:18.536 1396-1396/? I/Activity tag: Launching of main activity'. Annotations include: a green box around 'Activity tag:' with a green arrow pointing to 'Tag Id'; a purple box around 'Launching of main activity' with a purple arrow pointing to 'Message to display'; and a red box around 'MyFirstTag' with a red arrow pointing to 'Selected filter'.

logcat ADB logs → Memory → Log level: Verbose

09-02 19:07:18.536 1396-1396/? I/Activity tag: Launching of main activity

Tag Id

Message to display



# Webography

- ▶ [www.android.com](http://www.android.com)
- ▶ [developer.android.com](http://developer.android.com)
  - <http://developer.android.com/training/index.html>