

# Introduction to Android App Development

Hui Chen <sup>a</sup>

<sup>a</sup>CUNY Brooklyn College, Brooklyn, NY, USA

February 15, 2022

# Outline

- 1 Tools of the Trade
- 2 Android
- 3 IDE
- 4 Android SDK
- 5 Android Project
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production

# Outline

- 1 Tools of the Trade
- 2 Android
- 3 IDE
- 4 Android SDK
- 5 Android Project
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production

# Technical Readiness Preparation

Provide a technical readiness for

- ▶ Version Control Systems
- ▶ Develop simple Android Apps with Android Activities and Android Intent
- ▶ Test Android Apps

# Outline

- 1 Tools of the Trade
- 2 Android**
- 3 IDE
- 4 Android SDK
- 5 Android Project
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production

# Lecture Module Outline

- ▶ An introduction to Android
- ▶ The Android OS
- ▶ The IDE and build system
- ▶ Basic App Development

*Note: this is not an Android class!*

## Historical Context

- ▶ 2005 – Google, Inc. buys Android, Inc.
- ▶ 2007 – Google unveils Android OS
- ▶ 2008 – First mobile device to support Android
- ▶ ... many different versions of Android supporting many different devices (e.g. “IceCreamSandwich”, “Cupcake”, “KitKat”, “Lollipop”, “Marshmallow”, “Nougat”, “Oreo”, ..., “Tiramisu”)
  - ▶ Phone and Tablet
  - ▶ Wearables
  - ▶ Android TV
  - ▶ Automotives
- ▶ Check out Android version history from [Wikipedia](#)

## Device to OS Version Mapping

Keep in mind that targeting a newer OS may or may not be to your advantage – makes your app incompatible with a population of users



# The Android Software Stack

Let's have a look at  
[Android Software Stack](#)

# Outline

- 1 Tools of the Trade
- 2 Android
- 3 IDE**
- 4 Android SDK
- 5 Android Project
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production

# The IDE

Android Studio is the recommended IDE for app development

Have you used an IDE before, what is it?

Why IDE?

Why not IDE?

# Android Studio

- ▶ Improves productivity: edit, build, debug ... all in one place
- ▶ Replaced Eclipse-based development
- ▶ Based on the IntelliJ IDE
- ▶ Uses the Gradle build system, by default

```
1  buildscript {
2    repositories {
3      jcenter()
4    }
5
6    dependencies {
7      classpath 'com.android.tools.build:gradle:1.2.3'
8      // NOTE: Do not place your application dependencies here;
9      // they belong in the individual module build.gradle files
10   }
11   // ...
12   // ...
13 }
```

# Your Task: Download and Install Android Studio

<https://developer.android.com/studio>

# Outline

- 1 Tools of the Trade
- 2 Android
- 3 IDE
- 4 Android SDK**
- 5 Android Project
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production

# Android SDK

Android System Development Kit (SDK) for a specific version/release of the OS provides the development libraries

- ▶ Matching API versions to OS versions
- ▶ e.g. Android 5.0 = “Lollipop” = API 21
- ▶ Download while installing Android Studio Or later via SDK Manager

Documentation:

[Android API Reference](#)

# Outline

- 1 Tools of the Trade
- 2 Android
- 3 IDE
- 4 Android SDK
- 5 **Android Project**
  - Android Manifest File
  - Resource Files
  - Designing UI
  - Writing Basic Android Code
  - Running Apps
  - Release to Production



## Android Project in Android Studio

- ▶ Android Manifest file (an XML file)
  - ▶ UI definition in XML
  - ▶ Other definitions, e.g. strings for internationalization
- ▶ Java files (Your code)
- ▶ Gradle build scripts and settings
- ▶ Other resources (e.g. images, animations)

# Android Manifest File

All *activities*, *services* and *content provider* components of the application must be statically declared in this file

```
1 <manifest xmlns:android="http://schemas.android.com/apk/res/
  android" package="com.example.android.rssreader">
2 <uses-sdk android:minSdkVersion="16" android:targetSdkVersion="
  19" />
3 <uses-permission android:name="android.permission.INTERNET" />
4 <application android:name="RssApplication">
5 <activity android:name="RssfeedActivity" android:label="
  @string/title_activity_main" >
6   <intent-filter>
7     <action android:name="android.intent.action.MAIN" />
8     <category android:name="android.intent.category.LAUNCHER"
  />
9   </intent-filter>
10 </activity>
11 ...
12
```

# Content in Android Manifest File

A number of setting for the app:

- ▶ Target compiler and minimum API version
- ▶ Starting activities, intent filters
- ▶ Permission to use particular sensors, network or other capabilities
- ▶ Installation location
- ▶ ...

# Resource files

Lots of important stuff in the res folder

- ▶ Drawables – /res/drawable
- ▶ Simple Values – /res/values
- ▶ Layouts – /res/layout
- ▶ Raw Data – /res/raw
- ▶ Menus – /res/menu

# Designing the UI

## Responsive design

- ▶ Fit across wide variety of devices, and orientations

Use the graphical editor or just edit the XML

- ▶ Being able to read the generated XML helps

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:paddingBottom="@dimen/activity_vertical_margin"
4     android:paddingLeft="@dimen/activity_horizontal_margin"
5     android:paddingRight="@dimen/activity_horizontal_margin"
6     android:paddingTop="@dimen/activity_vertical_margin"
7     tools:context=".MainActivity" >
8 <TextView
9     android:id="@+id/mytext"
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:text="@string/hello_world" />
13 </RelativeLayout>
14 ...
15
```

# Typical Android UI elements

- ▶ Layouts
- ▶ Fragments
- ▶ Individual Widgets (e.g. Buttons, Views, EditTexts)
- ▶ Composite Widgets (e.g. ListView)

Each UI element has its own set of properties that govern its ID, look, and behavior

UI elements are typically defined in XML, but accessed and modified in Java

## Some Android Program Elements

- ▶ Activities – We'll deal with these most
- ▶ Services, i.e., UI-less elements
- ▶ Broadcast receivers, e.g., register to receive notification if someone is sending you a message
- ▶ Content providers – to encapsulate data

# Android Activity

```
1 package cisc3171.bc.cuny.edu;
2 import android.app.Activity;
3 import android.os.Bundle;
4 import android.widget.TextView;
5
6
7 // extends android.app.Activity
8 public class MainActivity extends Activity {
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13        TextView textView = (TextView) findViewById(R.id.mytext);
14        // ...
15    }
16 }
17
```



# Callbacks

Typically reliant on callback mechanisms

```
1 final Button button = (Button) findViewById(R.id.button_id);
2 button.setOnClickListener(
3     new View.OnClickListener() {
4         public void onClick(View v) {
5             // Perform action on click
6         }
7     }
8 );
9
```

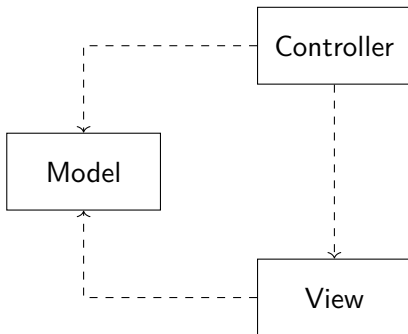
Link to the UI elements via their ID

Can programmatically set most element's properties

Recall: Anonymous classes, functional interface, lambda expression in Java

# Model-View-Controller

Generally follow the Model-View-Controller architecture, and classes are named after the components in the architecture, such as, `MainView`, `NavigationController`, and `SideFragmentController`.



## Making Apps from Android Projects

1. Java files compiled in class files, packaged into a .dex (Dalvik Executable) file
2. .dex files with resources (images, XML files) are packaged into an app bundle (an .aab file – replaces .apk file)
3. Since Android version 5.0 apps are executed by the Android Run-Time (ART), which pre-compiles the code to machine language.  
pre-compilation = larger executable files, faster runtime

# Running Apps during Development

On an Android device

- ▶ What version/release of Android on the device?
- ▶ Make sure the device is in development mode
- ▶ Settings → Development Options → USB Debugging (what if I don't see the Development Options?)
- ▶ Connect via USB cable
- ▶ Download drivers (?)

On the emulator (Android Virtual Device (AVD))

- ▶ Configure emulator to the version that you want to execute on
  - ▶ need a system image (big, i.e., 1 GB)
  - ▶ recommend to use x86 image for x86 hosts
- ▶ Don't close the emulator after executing, it may take time to start up

# Release the App

Google Play Store and ...

# References

- ▶ <https://developer.android.com/codelabs/build-your-first-android-app>
- ▶ Free courses (such as those on Udacity)
- ▶ <https://developer.android.com/>