Social Media Apps Programming

Developing Android Native Apps with Java (Android Studio)

1071SMAP03
TLMXM1A (8550) (M2143) (Fall 2018)
(MIS MBA) (2 Credits, Elective) [Full English Course]
Thu 8,9 (10:10-12:00) B206

Min-Yuh Day, Ph.D.
Assistant Professor

Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday

2018-09-27
# Course Schedule (1/2)

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2018/09/13</td>
<td>Course Orientation and Introduction to Social Media and Mobile Apps Programming</td>
</tr>
<tr>
<td>2</td>
<td>2018/09/20</td>
<td>Introduction to Android / iOS Apps Programming</td>
</tr>
<tr>
<td>3</td>
<td>2018/09/27</td>
<td>Developing Android Native Apps with Java (Android Studio)</td>
</tr>
<tr>
<td>4</td>
<td>2018/10/04</td>
<td>Developing iPhone / iPad Native Apps with Swift (XCode)</td>
</tr>
<tr>
<td>5</td>
<td>2018/10/11</td>
<td>Mobile Apps using HTML5/CSS3/JavaScript</td>
</tr>
<tr>
<td>6</td>
<td>2018/10/18</td>
<td>jQuery Mobile</td>
</tr>
<tr>
<td>7</td>
<td>2018/10/25</td>
<td>Create Hybrid Apps with Phonegap</td>
</tr>
<tr>
<td>8</td>
<td>2018/11/01</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>9</td>
<td>2018/11/08</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>2018/11/15</td>
<td>Midterm Exam Week / Project Presentation</td>
</tr>
<tr>
<td>11</td>
<td>2018/11/22</td>
<td>Case Study on Social Media Apps Programming and Marketing in Google Play and App Store</td>
</tr>
<tr>
<td>12</td>
<td>2018/11/29</td>
<td>Google Cloud Platform</td>
</tr>
<tr>
<td>13</td>
<td>2018/12/06</td>
<td>Google App Engine</td>
</tr>
<tr>
<td>14</td>
<td>2018/12/13</td>
<td>Google Map API</td>
</tr>
<tr>
<td>15</td>
<td>2018/12/20</td>
<td>Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)</td>
</tr>
<tr>
<td>16</td>
<td>2018/12/27</td>
<td>Twitter API</td>
</tr>
<tr>
<td>17</td>
<td>2019/01/03</td>
<td>Final Project Presentation</td>
</tr>
<tr>
<td>18</td>
<td>2019/01/10</td>
<td>Final Exam Week / Final Project Presentation</td>
</tr>
</tbody>
</table>
Android /iOS Apps Programming

Native Apps

Hybrid Apps

Mobile Web Apps
App Development Comparison

<table>
<thead>
<tr>
<th>Device Access</th>
<th>Speed</th>
<th>Development Cost</th>
<th>App Store</th>
<th>Approval Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native Apps</td>
<td>Full</td>
<td>Very Fast</td>
<td>Expensive</td>
<td>Available</td>
</tr>
<tr>
<td>Hybrid Apps</td>
<td>Full</td>
<td>Native Speed as Necessary</td>
<td>Reasonable</td>
<td>Available</td>
</tr>
<tr>
<td>Web Apps</td>
<td>Partial</td>
<td>Fast</td>
<td>Reasonable</td>
<td>Not Available</td>
</tr>
</tbody>
</table>

Outline

• Developing Android Native Apps with Java
  – Android Studio
  – Eclipse
  – Android Developer Tools (ADT) Bundle
  – Building Your First Android App
Native App Development

Android - Native App Development

Native App – Interaction with Mobile Device

Android

The new Pixel 2.
A more impressive camera and the Google Assistant built-in.

http://www.android.com/
Now available in the U.S. from Project Fi

Experience the best of Android on the all new Moto X4

LEARN MORE

http://www.android.com/
Android 8.0 Oreo is here

Smarter, faster, and more powerful than ever.

http://www.android.com/
## Android Platform

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3 -</td>
<td>Gingerbread</td>
<td>10</td>
<td>0.6%</td>
</tr>
<tr>
<td>2.3.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.0.3 -</td>
<td>Ice Cream</td>
<td>15</td>
<td>0.6%</td>
</tr>
<tr>
<td>4.0.4</td>
<td>Sandwich</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>2.3%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>3.3%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>1.0%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>14.5%</td>
</tr>
<tr>
<td>5.0</td>
<td>Lollipop</td>
<td>21</td>
<td>6.7%</td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td>22</td>
<td>21.0%</td>
</tr>
<tr>
<td>6.0</td>
<td>Marshmallow</td>
<td>23</td>
<td>32.0%</td>
</tr>
<tr>
<td>7.0</td>
<td>Nougat</td>
<td>24</td>
<td>15.8%</td>
</tr>
<tr>
<td>7.1</td>
<td></td>
<td>25</td>
<td>2.0%</td>
</tr>
<tr>
<td>8.0</td>
<td>Oreo</td>
<td>26</td>
<td>0.2%</td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on October 2, 2017.

http://developer.android.com/about/dashboards/index.html
Android Platform

Data collected during a 7-day period ending on October 2, 2017.

http://developer.android.com/about/dashboards/index.html
## Android Screen Sizes and Densities

<table>
<thead>
<tr>
<th></th>
<th>ldpi</th>
<th>mdpi</th>
<th>tvdpi</th>
<th>hdpi</th>
<th>xhdpi</th>
<th>xxhdpi</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>0.7%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.7%</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>1.5%</td>
<td>0.2%</td>
<td>30.5%</td>
<td>36.7%</td>
<td>21.3%</td>
<td>90.2%</td>
</tr>
<tr>
<td>Large</td>
<td>0.1%</td>
<td>2.9%</td>
<td>1.6%</td>
<td>0.5%</td>
<td>0.9%</td>
<td>0.1%</td>
<td>6.1%</td>
</tr>
<tr>
<td>Xlarge</td>
<td></td>
<td>2.0%</td>
<td></td>
<td>0.5%</td>
<td>0.5%</td>
<td></td>
<td>3.0%</td>
</tr>
<tr>
<td>Total</td>
<td>0.8%</td>
<td>6.4%</td>
<td>1.8%</td>
<td>31.5%</td>
<td>38.1%</td>
<td>21.4%</td>
<td></td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on October 2, 2017.

Android Development Environment

1. JDK
   (Java Development Kit)

2. Android Studio
1. JDK (Java Development Kit)

[Image showing Oracle website for JDK downloads]

Java SE Downloads

Java Platform (JDK) 9
NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 9
Java SE 9 is the latest update to the Java Platform. This release includes much awaited new features like the modularization of the Java Platform, better performance, support for new standards, and many other improvements.

Learn more

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Licensing Information User Manual
- Third Party Licenses
- Certified System Configurations
- Readme

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Developer Training
- Tutorials
- Java.com

Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
2.3.3 FOR MAC (463 MB)

› Read the docs    › See the release notes

Building Your First App

Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project with Android Studio and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, download and install Android Studio.

Source: https://developer.android.com/training/basics/firstapp/index.html
Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. Activity
   - activated by an asynchronous message
2. Service
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. Activity
   a single screen with a user interface

2. Service

3. Broadcast Receiver

4. Content Provider

Android App Building Blocks

1. Activity
2. Service — runs in the **background** to perform long-running operations or to perform work for remote processes
3. Broadcast Receiver
4. Content Provider

Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
   - responds to system-wide broadcast announcements
4. Content Provider

Android App Building Blocks

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

manages a shared set of application data
Developing Android Apps

1. **Screen Layout Design**: Views and Layouts
   - Graphical Layout
   - activity_main.xml

2. **App Components (Activity) Programming**
   - MainActivity.java
Building a Simple User Interface

- Create a Linear Layout
- Add a Text Field
- Add String Resources
- Add a Button
- Make the Input Box Fill in the Screen Width

Building a Simple User Interface

Building a Simple User Interface

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Android App Activity Lifecycle

Source:
The development process for Android applications

1. Setup

Set up your development environment

Install the Android SDK, Android Development Tools, and Android platforms.

Set up AVDs and devices for testing

Create Android Virtual Devices and connect hardware devices that will be used for testing.
2. Development

Create an Android project with your source code, resource files, and Android manifest file.
3. Debugging and Testing

1. Build and run your application in debug mode.
2. Debug your application using the Android debugging and logging tools.
3. Test your application using the Android testing and instrumentation framework.
4. Publishing

- **Prepare your application for release**
  - Configure, build, and test your application in release mode.

- **Release your application**
  - Publicize, sell, and distribute your application to users.
Demo:
Building Your First Android App with Android Studio
Java SE Development Kit (JDK)

Java SE Development Kit (JDK)

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:
- Java Developer Newsletter: From your Oracle account, select Subscriptions, expand Technology, and subscribe to Java.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 9 checksum

Java SE Development Kit 9
You must accept the Oracle Binary Code License Agreement for Java SE to download this software. Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

<table>
<thead>
<tr>
<th>Product / File Description</th>
<th>File Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>298.02 MB</td>
<td>jdk-9_linux-x64_bin.rpm</td>
</tr>
<tr>
<td>Linux</td>
<td>330.23 MB</td>
<td>jdk-9_linux-x64_bin.gz</td>
</tr>
<tr>
<td>macOS</td>
<td>371.64 MB</td>
<td>jdk-9_osx-x64_bin.dmg</td>
</tr>
<tr>
<td>Windows</td>
<td>358.69 MB</td>
<td>jdk-9_windows-x64_bin.exe</td>
</tr>
<tr>
<td>Solaris SPARC</td>
<td>207.05 MB</td>
<td>jdk-9_solaris-sparcv9_bin.tar.gz</td>
</tr>
</tbody>
</table>

Android Studio

The Official IDE for Android

Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
2.3.3 FOR MAC (463 MB)

› Read the docs  › See the release notes

Before downloading, you must agree to the following terms and conditions.

Terms and Conditions

This is the Android Software Development Kit License Agreement

1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which

I have read and agree with the above terms and conditions

DOWNLOAD ANDROID STUDIO FOR MAC
Install Android Studio

Setting up Android Studio takes just a few clicks. (You should have already downloaded Android Studio.)

To install Android Studio on your Mac, proceed as follows:

1. Launch the Android Studio DMG file.
2. Drag and drop Android Studio into the Applications folder, then launch Android Studio.
3. Select whether you want to import previous Android Studio settings, then click OK.
4. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

That's it! The following video shows each step of the recommended setup procedure.
Install Android Studio

android-studio-ide-162.4069837-mac.dmg

Disk Image - 486.1 MB
Install Android Studio
Android Studio
You can import your settings from a previous version of Studio.

- I want to import my settings from a custom location
  Specify config folder or installation home of the previous version of Studio:
  /Applications

- I do not have a previous version of Studio or I do not want to import my settings

OK
Welcome! This wizard will set up your development environment for Android Studio. Additionally, the wizard will help port existing Android apps into Android Studio or create a new Android application project.
Choose the type of setup you want for Android Studio:

- **Standard**
  
  Android Studio will be installed with the most common settings and options. Recommended for most users.

- **Custom**
  
  You can customize installation settings and components installed.
If you want to review or change any of your installation settings, click Previous.

**Current Settings:**

<table>
<thead>
<tr>
<th>Total Download Size:</th>
<th>899 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDK Components to Download:</td>
<td></td>
</tr>
<tr>
<td>Android Emulator</td>
<td>102 MB</td>
</tr>
<tr>
<td>Android SDK Build-Tools 26.0.2</td>
<td>51.3 MB</td>
</tr>
<tr>
<td>Android SDK Platform 26</td>
<td>60.7 MB</td>
</tr>
<tr>
<td>Android SDK Platform-Tools</td>
<td>7.5 MB</td>
</tr>
<tr>
<td>Android SDK Tools</td>
<td>98.2 MB</td>
</tr>
<tr>
<td>Android Support Repository</td>
<td>339 MB</td>
</tr>
<tr>
<td>Google Repository</td>
<td>205 MB</td>
</tr>
<tr>
<td>Intel x86 Emulator Accelerator (HAXM installer)</td>
<td>217 KB</td>
</tr>
<tr>
<td>SDK Patch Applier v4</td>
<td>1.74 MB</td>
</tr>
<tr>
<td>Sources for Android 26</td>
<td>33.5 MB</td>
</tr>
</tbody>
</table>
Download Status

- Installing Android SDK Platform-Tools in /Users/imyday/Library/Android/sdk/platform-tools
- "Install Android SDK Platform-Tools (revision: 26.0.1)" complete.
- "Install Android SDK Platform-Tools (revision: 26.0.1)" finished.
- Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)."
- Downloading: https://dl.google.com/android/repository/extras/intel/haxm-macosx_r6_2_1.zip
- https://dl.google.com/android/repository/extras/intel/haxm-macosx_r6_2_1.zip
- HAXM installation
- IntelHAXM_6.2.1.dmg
- Release Notes.txt
- silent_install.sh
- silent_install_readme.txt
- "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" ready.
- Finishing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)"
- Installing Intel x86 Emulator Accelerator (HAXM installer) in
  /Users/imyday/Library/Android/sdk/extras/intel/Hardware_Accelerated_Execution_Manager
- "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" complete.
- "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" finished.
- Preparing "Install Google Repository (revision: 58)."
- Downloading: https://dl.google.com/android/repository/google_m2repository_gms_v11_3_rc05_wear_2_0_5.zip
- https://dl.google.com/android/repository/google_m2repository_gms_v11_3_rc05_wear_2_0_5.zip
Downloading Components

```
android-8.0.0/templates/ic_launcher_ldpi.png
android-8.0.0/templates/ic_launcher_xhdpi.png

"Install Android SDK Platform 26 (revision: 2)" ready.
Finishing "Install Android SDK Platform 26 (revision: 2)"
Installing Android SDK Platform 26 in /Users/imyday/Library/Android/sdk/platforms/android-26
"Install Android SDK Platform 26 (revision: 2)" complete.
"Install Android SDK Platform 26 (revision: 2)" finished.
Parsing /Users/imyday/Library/Android/sdk/build-tools/26.0.2/package.xml
Parsing /Users/imyday/Library/Android/sdk/emulator/package.xml
Parsing /Users/imyday/Library/Android/sdk/extras/android/m2repository/package.xml
Parsing /Users/imyday/Library/Android/sdk/extras/google/m2repository/package.xml
Parsing /Users/imyday/Library/Android/sdk/extras/intel
   /Hardware_Accelerated_Execution_Manager/package.xml
Parsing /Users/imyday/Library/Android/sdk/patcher/v4/package.xml
Parsing /Users/imyday/Library/Android/sdk/platform-tools/package.xml
Parsing /Users/imyday/Library/Android/sdk/platforms/android-26/package.xml
Parsing /Users/imyday/Library/Android/sdk/sources/android-26/package.xml
Parsing /Users/imyday/Library/Android/sdk/tools/package.xml

Android SDK is up to date.
Running Intel® HAXM installer
Silent installation Pass!
```
Welcome to Android Studio

Android Studio
Version 2.3.3

🌟 Start a new Android Studio project
- Open an existing Android Studio project
- Check out project from Version Control
- Import project (Eclipse ADT, Gradle, etc.)
- Import an Android code sample

👀 Configure ➤ Get Help ➤
Configure your new project

Application name: MyFirstAndroidApp
Company domain: imyday.example.com
Package name: com.example.imyday.myfirstandroidapp
  Include C++ support
Project location: /Users/imyday/AndroidStudioProjects/MyFirstAndroidApp
Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

- **Phone and Tablet**
  - **Minimum SDK**: API 15: Android 4.0.3 (IceCreamSandwich)
    - Lower API levels target more devices, but have fewer features available.
    - By targeting API 15 and later, your app will run on approximately 100.0% of the devices that are active on the Google Play Store.
    - Help me choose

- **Wear**
  - **Minimum SDK**: API 21: Android 5.0 (Lollipop)

- **TV**
  - **Minimum SDK**: API 21: Android 5.0 (Lollipop)

- **Android Auto**
Select the form factors your app will run on

Different platforms may require separate SDKs

- **Phone and Tablet**
  - Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)
  - API 15: Android 4.0.3 (IceCreamSandwich)
  - API 16: Android 4.1 (Jelly Bean)
  - API 17: Android 4.2 (Jelly Bean)
  - API 18: Android 4.3 (Jelly Bean)
  - API 19: Android 4.4 (KitKat)
  - API 20: Android 4.4W (KitKat Wear)
  - API 21: Android 5.0 (Lollipop)
  - API 22: Android 5.1 (Lollipop)

- **Wear**
- **TV**

- **Minimum SDK**
  - API 21: Android 5.0 (Lollipop)
Select the form factors your app will run on

Different platforms may require separate SDKs

- Phone and Tablet
  - Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)
  - API 15: Android 4.0.3 (IceCreamSandwich)
  - API 16: Android 4.1 (Jelly Bean)
  - API 17: Android 4.2 (Jelly Bean)
  - API 18: Android 4.3 (Jelly Bean)
  - API 19: Android 4.4 (KitKat)
  - API 20: Android 4.4W (KitKat Wear)
  - API 21: Android 5.0 (Lollipop)
  - API 22: Android 5.1 (Lollipop)

- Wear
  - Minimum SDK: API 21: Android 5.0 (Lollipop)

- TV
  - Minimum SDK

- Android Auto
Installing Requested Components

SDK Path: /Users/imyday/Library/Android/sdk

constraintLayout 1.0.2 (revision: 1) "layout 1.0.2 in /Users/imyday/Library/Android/sdk/extras/m2repository/com/android/support/constraint/constraint-layout-solver/1.0.2
layout 1.0.2 (revision: 1)" complete.
yout for Android 1.0.2 (revision: 1)"
com/android/repository/com.android.support.constraint-constraint-layout-1.0.2.zip
ndroid 1.0.2 (revision: 1)" ready.
yout for Android 1.0.2 (revision: 1)"
Android 1.0.2 in /Users/imyday/Library/Android/sdk/extras/m2repository/com/android/support/constraint/constraint-layout/1.0.2
ndroid 1.0.2 (revision: 1)" complete.
yndroid 1.0.2 (revision: 1)" finished.
ndroid/sdk/build-tools/26.0.2/package.xml
ndroid/sdk/emulator/package.xml
ndroid/sdk/extras/android/m2repository/package.xml
ndroid/sdk/extras/google/m2repository/package.xml
ndroid/sdk/extras/intel/Hardware_Accelerated_Execution_Manager/package.xml
ndroid/sdk/extras/m2repository/com/android/support/constraint/constraint-layout-solver/1.0.2/package.xml
ndroid/sdk/extras/m2repository/com/android/support/constraint/constraint-layout/1.0.2/package.xml
ndroid/sdk/patcher/v4/package.xml
ndroid/sdk/platform-tools/package.xml
ndroid/sdk/platforms/android-26/package.xml
ndroid/sdk/sources/android-26/package.xml

Done
Add an Activity to Mobile

- Add No Activity
- Basic Activity
- Bottom Navigation Activity
- Empty Activity
- Fullscreen Activity
- Google AdMob Ads Activity
- Google Maps Activity
- Login Activity
- Master/Detail Flow
- Navigation Drawer Activity
- Scrolling Activity
- Settings Activity

Create New Project

[Next] [Finish] [Previous] [Cancel]
Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

Generate Layout File

Layout Name: activity_main

Backwards Compatibility (AppCompat)

Empty Activity

The name of the activity class to create
package com.example.imyday.myfirstandroidapp;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
package com.example.imyday.myfirstandroidapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.
### Choose a device definition

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Play Store</th>
<th>Size</th>
<th>Resolution</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV</td>
<td>Pixel XL</td>
<td></td>
<td>5.5&quot;</td>
<td>1440x2...</td>
<td>560dpi</td>
</tr>
<tr>
<td></td>
<td>Pixel</td>
<td></td>
<td>5.0&quot;</td>
<td>1080x1...</td>
<td>xxhdpi</td>
</tr>
<tr>
<td></td>
<td>Nexus 5</td>
<td></td>
<td>4.0&quot;</td>
<td>480x800</td>
<td>hdpi</td>
</tr>
<tr>
<td></td>
<td>Nexus One</td>
<td></td>
<td>3.7&quot;</td>
<td>480x800</td>
<td>hdpi</td>
</tr>
<tr>
<td></td>
<td>Nexus 6P</td>
<td></td>
<td>5.7&quot;</td>
<td>1440x2...</td>
<td>560dpi</td>
</tr>
<tr>
<td></td>
<td>Nexus 6</td>
<td></td>
<td>5.96&quot;</td>
<td>1440x2...</td>
<td>560dpi</td>
</tr>
<tr>
<td>Phone</td>
<td>Nexus 5X</td>
<td></td>
<td>5.2&quot;</td>
<td>1080x1...</td>
<td>420dpi</td>
</tr>
<tr>
<td></td>
<td>Nexus 5</td>
<td></td>
<td>4.95&quot;</td>
<td>1080x1...</td>
<td>xxhdpi</td>
</tr>
<tr>
<td></td>
<td>Nexus 4</td>
<td></td>
<td>4.7&quot;</td>
<td>768x12...</td>
<td>xhdpi</td>
</tr>
<tr>
<td></td>
<td>Galaxy Nexus</td>
<td></td>
<td>4.65&quot;</td>
<td>720x12...</td>
<td>xhdpi</td>
</tr>
<tr>
<td></td>
<td>5.4&quot; FWVGA</td>
<td></td>
<td>5.4&quot;</td>
<td>480x854</td>
<td>mdpi</td>
</tr>
<tr>
<td></td>
<td>5.1&quot; WVGA</td>
<td></td>
<td>5.1&quot;</td>
<td>480x800</td>
<td>mdpi</td>
</tr>
</tbody>
</table>

**Nexus 5X**

- Size: large
- Ratio: long
- Density: 420dpi

**Device Diagram**

- Size: 1080px
- Ratio: 1920px
- Width: 5.2"
A system image must be selected to continue.
License Agreement

Android Studio

To get started with the Android SDK Preview, you must agree to the following terms and conditions. As described below, please note that this is a preview version of the Android SDK, subject to change, that you use at your own risk. The Android SDK Preview is not a stable release, and may contain errors and defects that can result in serious damage to your computer systems, devices and data.

This is the Android SDK Preview License Agreement (the "License Agreement").

1. Introduction

1.1 The Android SDK Preview (referred to in the License Agreement as the "Preview" and specifically including the Android system files, packaged APIs, and Preview library files, if and when they are made available) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the Preview.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: http://source.android.com/, as updated from time to time.

1.3 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

2. Accepting the License Agreement

After reading the License Agreement, finish installation by choosing Accept.
Verify Configuration

AVD Name: Nexus 5X API 26

- [ ] Nexus 5X
- [ ] Android 8.0 x86

Startup orientation:
- Portrait
- Landscape

Default Orientation

Sets the initial orientation of the device. During AVD emulation you can also rotate the device screen.

Emulated Performance Graphics: Automatic

Device Frame: Enable Device Frame

Show Advanced Settings

Options:
- Cancel
- Previous
- Next
- Finish
<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Play Store</th>
<th>Resolution</th>
<th>API</th>
<th>Target</th>
<th>CPU/ABI</th>
<th>Size on Disk</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nexus 5X API 26</td>
<td>📝</td>
<td>1080 x 1920: 420dpi</td>
<td>26</td>
<td>Android 8.0 (Google ...</td>
<td>x86</td>
<td>650 MB</td>
<td></td>
</tr>
</tbody>
</table>
package com.example.imyday.myfirstandroidapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.example.imyday.myfirstandroidapp">
  <application>
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
Hello World
Demo: Building Your First Android App with Android Developer Tools (ADT)
Building Your First App

Welcome to Android application development!

This class teaches you how to build your first Android app. You'll learn how to create an Android project and run a debuggable version of the app. You'll also learn some fundamentals of Android app design, including how to build a simple user interface and handle user input.

Before you start this class, be sure you have your development environment set up. You need to:

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
3. Download the latest SDK tools and platforms using the SDK Manager.

If you haven't already done these tasks, start by downloading the Android SDK and following the install steps. Once you've finished the setup, you're ready to begin this class.

This class uses a tutorial format that incrementally builds a small Android app that teaches you some fundamental concepts about Android development, so it's important that you follow each step.

Start the first lesson

Except as noted, this content is licensed under Creative Commons Attribution 2.5. For details and restrictions, see the Content License.

http://developer.android.com/training/basics/firstapp/index.html
Creating an Android Project

An Android project contains all the files that comprise the source code for your Android app. The Android SDK tools make it easy to start a new Android project with a set of default project directories and files.

This lesson shows how to create a new project either using Eclipse (with the ADT plugin) or using the SDK tools from a command line.

**Note:** You should already have the Android SDK installed, and if you’re using Eclipse, you should also have the ADT plugin installed (version 21.0.0 or higher). If you don’t have these, follow the guide to Installing the Android SDK before you start this lesson.

Create a Project with Eclipse

1. Click **New** in the toolbar.
2. In the window that appears, open the Android folder, select **Android Application Project**, and click **Next**.
3. Fill in the form that appears:
   - **Application Name** is the app name that appears to users. For this project, use “My First App.”
   - **Project Name** is the name of your project directory and the name visible in Eclipse.
Create Activity
Select whether to create an activity, and if so, what kind of activity.

- Create Activity
- Blank Activity
- Fullscreen Activity
- Master/Detail Flow

Blank Activity
Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```java
package tw.edu.tku.im.sm2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public void onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
    }
}
```
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

Auto Monitor Logcat

Would you like ADT to automatically monitor logcat output for messages from applications in the workspace?

- No, do not monitor logcat output.
- Yes, monitor logcat and display logcat view if there are messages with priority higher than:

  error

OK
Getting Started

Building Your First App
- Creating an Android Project
- Running Your Application
- Building a Simple User Interface

Starting Another Activity
- Adding the Action Bar
- Supporting Different Devices
- Managing the Activity Lifecycle
- Building a Dynamic UI with Fragments
- Saving Data
- Interacting with Other Apps

Building Apps with Content Sharing
Building Apps with Multimedia
Building Apps with Graphics & Animation
Building Apps with Connectivity & the Cloud
Building Apps with User Info & Location

- Best Practices for

Figure 2. Both activities in the final app, running on Android 4.0.

That's it, you've built your first Android app!

To learn more, follow the link below to the next class.

http://developer.android.com/training/basics/firstapp/starting-activity.html
public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myFirstApp";

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public void sendMesage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_text);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
Hello World Myday
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="horizontal">
   <EditText android:id="@+id/edit_message"
      android:layout_weight="1"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:hint="@string/edit_message" />
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/button_send"
      android:onClick="sendMessage" />
</LinearLayout>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".DisplayMessageActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">My First App</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="edit_message">Enter a message</string>
    <string name="button_send">Send</string>
    <string name="title_activity_main">MainActivity</string>
    <string name="title_activity_display_message">My Message</string>

</resources>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="tw.edu.tku.im.smap2013.imyday.myfirstapp"
  android:versionCode="1"
  android:versionName="1.0">
  <uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="18"/>
  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
      android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.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>
    <activity
      android:name="tw.edu.tku.im.smap2013.imyday.myfirstapp.DisplayMessageActivity"
      android:label="@string/title_activity_display_message"
      android:parentActivityName="tw.edu.tku.im.smap2013.imyday.myfirstapp">
      <meta-data
        android:name="android.support.PARENT_ACTIVITY"
        android:value="tw.edu.tku.im.smap2013.imyday.myfirstapp"/>
    </activity>
  </application>
</manifest>
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.MenuItem;
import android.support.v4.app.NavUtils;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.os.Build;
import android.content.Intent;
import android.widget.TextView;

public class DisplayMessageActivity extends Activity {

    @SuppressLint("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_message);
        // Show the Up button in the action bar.
        setupActionBar();

        // Get the message from the intent
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

        // Great the text view
        TextView textView = new TextView(this);
        textView.setTextSize(40);
        textView.setText(message);

        setContentView(textView);

    }

    ...
}
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.content.Intent;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myfirstapp.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    /** Called when the user clicks the Send button */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
/**
 * Set up the [@link android.app.ActionBar](https://developer.android.com/reference/android/app/ActionBar.html), if the API is available.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void setupActionBar() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.display_message, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case android.R.id.home:
        // This ID represents the Home or Up button. In the case of this
        // activity, the Up button is shown. Use NavUtils to allow users
        // to navigate up one level in the application structure. For
        // more details, see the Navigation pattern on Android Design:
        //
        // http://developer.android.com/design/patterns/navigation.html#up-vs-back
        //
        NavUtils.navigateUpFromSameTask(this);
        return true;
    }

    return super.onOptionsItemSelected(item);
}
Alternatives for Developing Android Apps

- MIT App Inventor
  - drag-and-drop Android app development tool.
  - [http://appinventor.mit.edu/](http://appinventor.mit.edu/)

- Appery.io
  - develop apps for Android (iOS / Windows Phone).
  - [http://appery.io](http://appery.io)

- Appnotch
  - drag-and-drop service that allows you to develop apps for Android (iOS).
MIT App Inventor

http://appinventor.mit.edu/
Accelerate Mobile Innovation in the Enterprise

The only cloud-based platform with visual development tools and integrated backend services

Try Our Free Plan

http://appery.io/
Easiest way to create Apps.

- Build HTML5, iOS and Android apps.
- Just drag & drop. No coding required.
- Nothing to install. Build on Cloud.
- Get your live app in 3 easy steps.

Start Building for Free

Reasons AppNotch is getting popular

Transform your Ideas to Apps!
Build interactive mockups, wireframes and clickable working prototypes in HTML5, Android, iPhone and iPad.

Did we say this is 100% free?

Start Prototyping

Beautiful, Flexible and Powerful!
Create top-notch apps with drag & drop templates, themes, widgets, controls, forms with instant hosting.

Did we say this is affordably priced?

Start Exploring

Personal Tour and Private Demo!
We'll show you an addictively easy way to create beautiful web, native and hybrid apps, without coding.

Did we say you will love the magic?

Schedule a Demo

http://www.appnotch.com/
References

- Android Developer: http://developer.android.com/
- Bill Butterfield (2017), Android Studio Tutorials, https://www.youtube.com/watch?v=dFIPARW5IX8&list=PLp9HFLVct_ZvMa7IVdQyUUyh8t2re9apm