Social Media Apps Programming

Developing Android Native Apps with Java (Eclipse)(MIT App Inventor)

1031SMAP03
TLMXM1A (8687) (M2143) (Fall 2014)
(MIS MBA) (2 Credits, Elective) [Full English Course]
Thu 8,9 (15:10-17:00) V201

Min-Yuh Day, Ph.D.
Assistant Professor
Department of Information Management
Tamkang University

http://mail.tku.edu.tw/myday
## Course Schedule (1/3)

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2014/09/17</td>
<td>Course Orientation and Introduction to Social Media and Mobile Apps Programming</td>
</tr>
<tr>
<td>2</td>
<td>2014/09/24</td>
<td>Introduction to Android / iOS Apps Programming</td>
</tr>
<tr>
<td>3</td>
<td>2014/10/01</td>
<td>Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)</td>
</tr>
<tr>
<td>4</td>
<td>2014/10/08</td>
<td>Developing iPhone / iPad Native Apps with Swift / Objective-C (XCode)</td>
</tr>
<tr>
<td>5</td>
<td>2014/10/15</td>
<td>Mobile Apps using HTML5/CSS3/JavaScript</td>
</tr>
<tr>
<td>6</td>
<td>2014/10/22</td>
<td>jQuery Mobile</td>
</tr>
</tbody>
</table>
# Course Schedule (2/3)

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Subject/Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 7</td>
<td>2014/10/29</td>
<td>Create Hybrid Apps with Phonegap</td>
</tr>
<tr>
<td>• 8</td>
<td>2014/11/05</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>• 9</td>
<td>2014/11/12</td>
<td>jQuery Mobile/Phonegap</td>
</tr>
<tr>
<td>• 10</td>
<td>2014/11/19</td>
<td>Midterm Exam Week (Midterm Project Report)</td>
</tr>
<tr>
<td>• 11</td>
<td>2014/11/26</td>
<td>Case Study on Social Media Apps Programming and Marketing in Google Play and App Store</td>
</tr>
<tr>
<td>• 12</td>
<td>2014/12/03</td>
<td>Google Cloud Platform</td>
</tr>
<tr>
<td>Week</td>
<td>Date</td>
<td>Subject/Topics</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>13</td>
<td>2014/12/10</td>
<td>Google App Engine</td>
</tr>
<tr>
<td>14</td>
<td>2014/12/17</td>
<td>Google Map API</td>
</tr>
<tr>
<td>15</td>
<td>2014/12/24</td>
<td>Facebook API (Facebook JavaScript SDK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Integrate Facebook with iOS/Android Apps)</td>
</tr>
<tr>
<td>16</td>
<td>2014/01/31</td>
<td>Twitter API</td>
</tr>
<tr>
<td>17</td>
<td>2015/01/07</td>
<td>Final Project Presentation</td>
</tr>
<tr>
<td>18</td>
<td>2015/01/14</td>
<td>Final Exam Week (Final Project Report)</td>
</tr>
</tbody>
</table>
Outline

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

• MIT App Inventor
Android /iOS Apps Programming

Native Apps

Hybrid Apps

Mobile Web Apps
Native App Development

Android - Native App Development

Native App – Interaction with Mobile Device

Coming to watches, phones, tablets, TVs, and cars near you

Android Wear  ›
Phones and Tablets  ›
Android TV  ›
Android Auto  ›

http://www.android.com/
Android 2.2, Froyo

Voice Typing lets you input text, and Voice Actions let you control your phone, just by speaking.
Android 2.3, Gingerbread

New sensors make Android great for gaming - so you can touch, tap, tilt, and play away.
Android 3.0, Honeycomb

Optimized for tablets, this release opens up new horizons wherever you are.
Android 4.0, Ice Cream Sandwich

Android comes of age with a new, refined design. Simple, beautiful and beyond smart.
Android 4.1, Jelly Bean

Android is fast and smooth with buttery graphics. With Google Now, you get just the right information at the right time.

And with more than 1 million apps on Google Play, and thousands of Android devices, you've got the freedom to do what you want on any device you choose.
Android 4.4, KitKat

It's our goal with Android KitKat to make an amazing Android experience available for everybody.
# Android Platform

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Froyo</td>
<td>8</td>
<td>0.7%</td>
</tr>
<tr>
<td>2.3.3 - 2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>11.4%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>9.6%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>25.1%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>20.7%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>8.0%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>24.5%</td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on September 9, 2014. Any versions with less than 0.1% distribution are not shown.

Android Platform

Data collected during a 7-day period ending on September 9, 2014.
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>6.2%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.2%</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
<td>10.6%</td>
<td></td>
<td>35.7%</td>
<td>19.2%</td>
<td>16.2%</td>
<td>81.7%</td>
</tr>
<tr>
<td>Large</td>
<td>0.5%</td>
<td>4.3%</td>
<td>1.7%</td>
<td>0.6%</td>
<td>0.6%</td>
<td></td>
<td>7.7%</td>
</tr>
<tr>
<td>Xlarge</td>
<td></td>
<td>3.7%</td>
<td></td>
<td>0.3%</td>
<td>0.4%</td>
<td></td>
<td>4.4%</td>
</tr>
<tr>
<td>Total</td>
<td>6.7%</td>
<td>18.6%</td>
<td>1.7%</td>
<td>36.6%</td>
<td>20.2%</td>
<td>16.2%</td>
<td></td>
</tr>
</tbody>
</table>

Data collected during a 7-day period ending on September 9, 2014.

Android Screen Sizes and Densities

![Diagram showing screen sizes and densities](http://developer.android.com/guide/practices/screens_support.html)
Android Development Environment

1. **JDK**
   (Java Development Kit)
   
   [Link](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

2. **ADT Bundle**
   (Android Developer Tools Bundle)
   (Eclipse + ADT plugin + Android SDK + Android Platform + emulator)
   
   [Link](http://developer.android.com/sdk/index.html)
1. JDK (Java Development Kit)

ADT Bundle

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

With a single download, the Eclipse ADT bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- A version of the Android platform
- A version of the Android system image for the emulator

Get Android Studio Beta

Android Studio is a new IDE powered by IntelliJ that provides new features and improvements over ADT. It's currently in beta but will be the official Android IDE once it's ready.

If you're a new Android developer, you should consider starting with Android Studio, because the ADT plugin for Eclipse is no longer in development.
Option 1. Eclipse ADT Bundle
Option 2. Android Studio
Building Your First Android App

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse
3. Download the latest SDK tools and platforms using the SDK Manager.

ADT Bundle

(Android Developer Tools Bundle)
(Eclipse + ADT plugin + Android SDK + Android Platform + emulator)

ADT Bundle

ADT (Android Developer Tools)

- With a single download, the ADT Bundle includes everything you need to begin developing apps:
  1. Eclipse + ADT plugin
  2. Android SDK Tools
  3. Android Platform-tools
  4. The latest Android platform
  5. The latest Android system image for the emulator

http://developer.android.com/training/basics/firstapp/index.html
Android Developer Tools

Build: v22.2.1-833290

This product includes Eclipse Platform, JDT, CDT, EMF, GEF and WTP, all of which are Copyright (c) Eclipse contributors and others. Visit http://eclipse.org/

Android Developer Tools are Copyright (c) The Android Open Source Project. Visit http://developer.android.com
Get the Android SDK

Before installing the Android SDK, 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 this 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 this License Agreement. This 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 "Google" means Google Inc., a Delaware corporation with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States.

I have read and agree with the above terms and conditions

Download Eclipse ADT with the Android SDK for Mac

https://dl.google.com/android/adt/adt-bundle-mac-x86_64-20140702.zip
Installing the Eclipse ADT Bundle

1. Unpack the ZIP file (named `adt-bundle-<os_platform>.zip`) and save it to an appropriate location, such as a "Development" directory in your home directory.

2. Open the `adt-bundle-<os_platform>/eclipse` directory and launch Eclipse.

Get Android Studio Beta

• **Android Studio** is a new IDE powered by IntelliJ that provides new features and improvements over ADT. It's currently in beta but will be the official Android IDE once it's ready.

• If you're a new Android developer, you should consider starting with Android Studio, because the **ADT plugin for Eclipse** is no longer in active development.

Installing the Eclipse ADT Bundle

The Eclipse ADT Bundle provides everything you need to start developing apps, including the Android SDK tools and a version of the Eclipse IDE with built-in ADT (Android Developer Tools) to streamline your Android app development.

If you didn't download the Eclipse ADT bundle, go download the Eclipse ADT bundle now, or switch to the Android Studio install or stand-alone SDK Tools install instructions.

To set up the ADT Bundle:

1. Unpack the ZIP file (named `adt-bundle-<os_platform>.zip`) and save it to an appropriate location, such as a "Development" directory in your home directory.
2. Open the `adt-bundle-<os_platform>/eclipse/` directory and launch Eclipse.

Caution: Do not move any of the files or directories from the `adt-bundle-<os_platform>` directory. If you move the `eclipse/` or `sdk/` directory, ADT will not be able to locate the SDK and you'll need to manually update the ADT preferences.

Eclipse with ADT is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

CONTINUE: ADDING SDK PACKAGES >

Adding SDK Packages

By default, the Android SDK does not include everything you need to start developing. The SDK separates tools, platforms, and other components into packages you can download as needed using the Android SDK Manager. So before you can start, there are a few packages you should add to your Android SDK.

To start adding packages, launch the Android SDK Manager in one of the following ways:

- In Eclipse or Android Studio, click SDK Manager in the toolbar.
- If you're not using Eclipse or Android Studio:
  - Windows: Double-click the SDK Manager.exe file at the root of the Android SDK directory.
  - Mac/Linux: Open a terminal and navigate to the tools directory in the Android SDK, then execute `android sdk`.

When you open the SDK Manager for the first time, several packages will be selected by default. Leave these selected, but be sure you have everything you need to get started by following these steps:

1. Get the latest SDK tools

   As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:
   1. Open the Tools directory and select:
      - Android SDK Tools
      - Android SDK Platform-tools
      - Android SDK Build-tools (highest version)
   2. Open the first Android X.X folder (the latest version) and select:

1. Get the latest SDK tools

As a minimum when setting up the Android SDK, you should download the latest tools and Android platform:

1. Open the Tools directory and select:
   - Android SDK Tools
   - Android SDK Platform-tools
   - Android SDK Build-tools (highest version)
2. Open the first Android X.X folder (the latest version) and select:
   - SDK Platform
   - A system image for the emulator, such as ARM EABI v7a System Image

2. Get the support library for additional APIs

The Android Support Library provides an extended set of APIs that are compatible with most versions of Android.

Open the Extras directory and select:
- Android Support Repository
- Android Support Library

The support library is required for:
- Android Wear
- Android TV
- Google Cast

It also provides these popular APIs:
- Navigation drawer
- Swipe views
- Backward-compatible action bar

3. Get Google Play services for even more APIs

To develop with Google APIs, you need the Google Play services package:

Open the **Extras** directory and select:

- **Google Repository**
- **Google Play services**

**Note:** Google Play services APIs are not available on all Android-powered devices, but are available on all devices with Google Play Store. To use these APIs in the Android emulator, you must also install the the **Google APIs** system image from the latest Android X.X directory in the SDK Manager.

The Google Play services APIs provide a variety of features and services for your Android apps, such as:

- User authentication
- Google Maps
- Google Cast
- Games achievements and leaderboards
- And much more

4. Install the packages

Once you've selected all the desired packages, continue to install:

1. Click **Install X packages**.
2. In the next window, double-click each package name on the left to accept the license agreement for each.
3. Click **Install**.

The download progress is shown at the bottom of the SDK Manager window. **Do not exit the SDK Manager** or it will cancel the download.

5. **Build something!**

With the above packages now in your Android SDK, you're ready to build apps for Android. As new tools and other APIs become available, simply launch the SDK Manager to download the new packages for your SDK.

Here are a few options for how you should proceed:

### Get started
If you're new to Android development, learn the basics of Android apps by following the guide to [Building Your First App](http://developer.android.com/sdk/installing/adding-packages.html).

### Build for wearables
If you're ready to start building apps for Android wearables, see the guide to [Building Apps for Android Wear](http://developer.android.com/sdk/installing/adding-packages.html).

### Use Google APIs
To start using Google APIs, such as Maps or Play Game services, see the guide to [Setting Up Google Play Services](http://developer.android.com/sdk/installing/adding-packages.html).
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.

Note: Make sure you install the most recent versions of the ADT plugin and the Android SDK before you start this class. The procedures described in this class may not apply to earlier versions.

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 →
Android App Building Blocks

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

Android App Building Blocks

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

activated by an asynchronous message

Android App Building Blocks

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

a single screen with a user interface

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

Source: http://developer.android.com/training/basics/firstapp/building-ui.html
Building a Simple User Interface

Building a Simple User Interface

Android App Activity Lifecycle

Demo: Building Your First Android App with Android Developer Tools (ADT)
public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "tw.edu.tku.im.smap2013.imyday.myFirstApp"

    @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
        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().setAction(Intent.ACTION_SEND);
        EditText editText = (EditText) findViewById(R.id.edit_text);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
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 »

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.
Welcome!
The Android Developer Tools provide a first-class dev integrated development environment is set up with the image so you can immediately begin building apps an

Tutorials
Build Your First App
If you're new to Android, follow that responds to input.
Design Your App
Before you begin developing from your app.
Test Your App
The Android framework provic expected under various concide

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.
Creating a new Android Application

Application Name: My First App
Project Name: MyFirstApp
Package Name: tw.edu.tku.im.smap2013.imyday.myfirstapp
Minimum Required SDK: API 18: Android 2.2 (Froyo)
Target SDK: API 18: Android 4.3 (Jelly Bean)
Compile With: API 18: Android 4.3 (Jelly Bean)
Theme: Holo Light with Dark Action Bar

The package name must be a unique identifier for your application.
It is typically not shown to users, but it *must* stay the same for the lifetime of your application; it is how multiple versions of the same application are considered the 'same app'.
This is typically the reverse domain name of your organization plus one or more application identifiers, and it must be a valid Java package name.
New Android Application

Configure Project

- Create custom launcher icon
- Create activity
- Mark this project as a library
- Create Project in Workspace

Location: /Users/imyday/Development/AndroidWorkspace/MyFirstApp

Working sets

- Add project to working sets

Working sets: 

Next >  
Finish
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.
Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name: MainActivity
Layout Name: activity_main
Navigation Type: None

The name of the activity class to create
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;
    }
}
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;
    }
}
package tw.edu.tku.im.smap2013.imyday.myfirstapp;

Select a device with min API level 8.

Receive a running Android device

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>AVD Name</th>
<th>Target</th>
<th>Debug State</th>
</tr>
</thead>
</table>

Launch a new Android Virtual Device

AVD Name | Target Name | Platform | API Level | CPU/ABI |
--- | --- | --- | --- | --- |
--- | No AVD available | --- | --- | --- |

Use same device for future launches

An Android Virtual Device that failed to load. Click 'Details' to see the error.

Android

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:

OK
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.menu_main, menu);
    return true;
}
}
on the second activity.

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
        // the device supports its implementation.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DisplayMessageActivity.class);
        String message = editText.setText().toString().toBeString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}

Hello World Myday
Hello World Myday
activity_main.xml

<?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>
strings.xml

<?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: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);

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

    ...
}
MainActivity.java

```java
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}, 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
  – http://appinventor.mit.edu/

• Appery.io
  – develop apps for Android (iOS / Windows Phone).
  – http://appery.io

• Appnotch
  – drag-and-drop service that allows you to develop apps for Android (iOS).
  – http://www.appnotch.com/
MIT App Inventor

Get Started
Follow these simple steps to build your first app.

Start

Teach
Teachers, find out about curriculum and teaching

Tutorials
Step-by-step guides show you how to build all kinds of apps.

Tutorials

Forums
Join community forums to get support and advice.

Invent Now

Weekly Active Users: 87K  Total Registered Users: 1.9M  Countries: 195  Apps Built: 4.7M

Recent News
AUGUST: Congratulations to co-App of the Month: Sparki Bluetooth Controller
AUGUST: Congratulations to co-App of the Month: Let Me Hear Again PRO
JULY: Congratulations to co-App of the Month: KidRescue!
The creation of live programming in 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 topnotch 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/
• MIT App Inventor: http://appinventor.mit.edu/
• Native, Web or Hybrid Mobile Apps?, https://www.youtube.com/watch?v=Ns-JS4amlTc