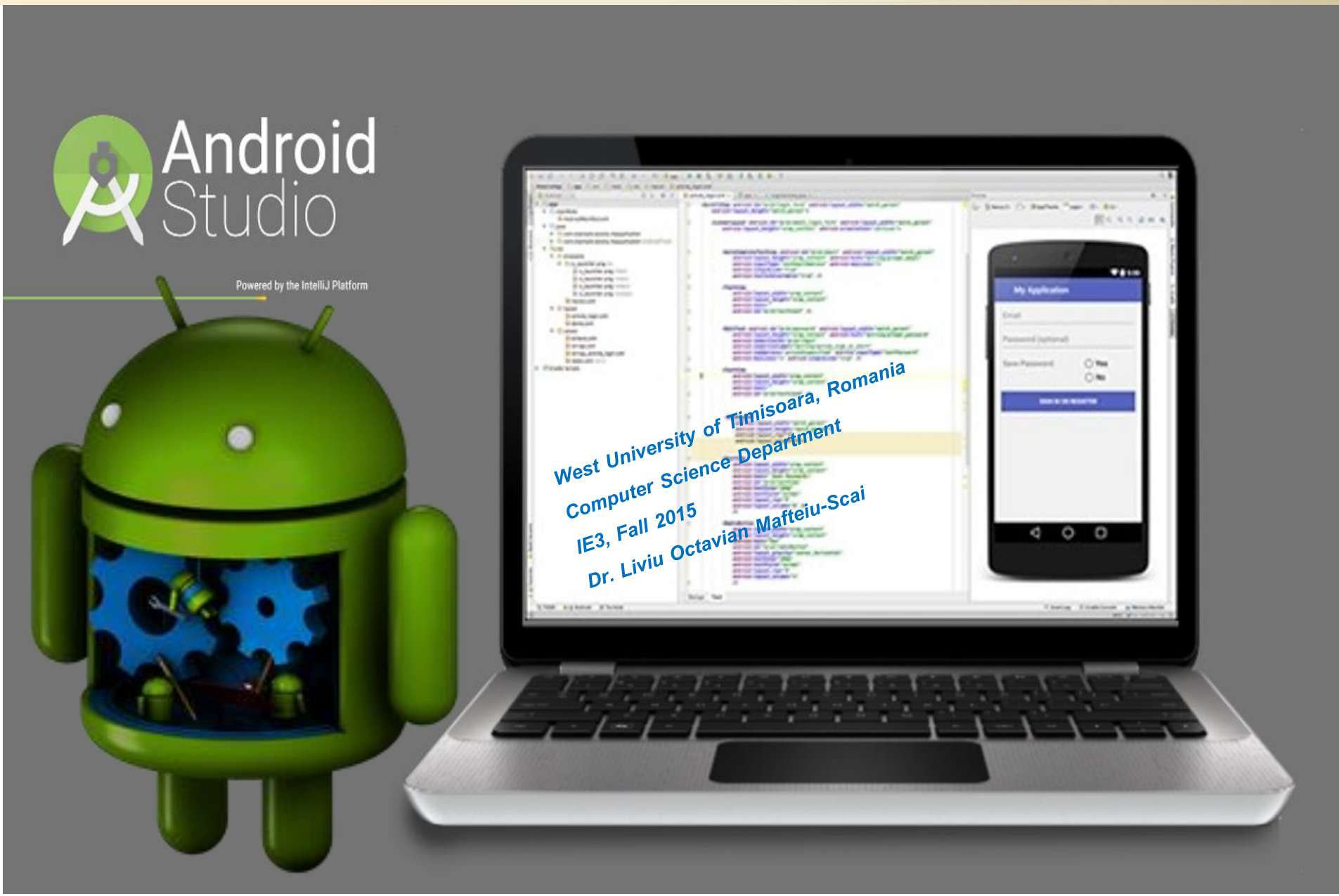
Mobile Application Development

Course 6



What is Android Studio?

Android Studio is the official IDE for Android application development, based on *IntelliJ IDEA*

- replace Eclipse Android Development Tools (ADT)
  -first Google's IDE for native Android apps develop.
- ver. 0.1 May 2013, ver. 0.8 released in June 2014.
  The first stable version: 1.0 released in Dec. 2014
  Curent: 1.4 sept 2015
- available for Windows, Mac OS X and Linux.

IntelliJ IDEA:
a Java IDE developed by JetBrains
-known as IntelliJ;
-ver. 1 ian. 2001, 14.1 sept. 2015;
-other software based on it:
AppCode, PhpStorm, PyCharm, RubyMine, WebStorm, MPS

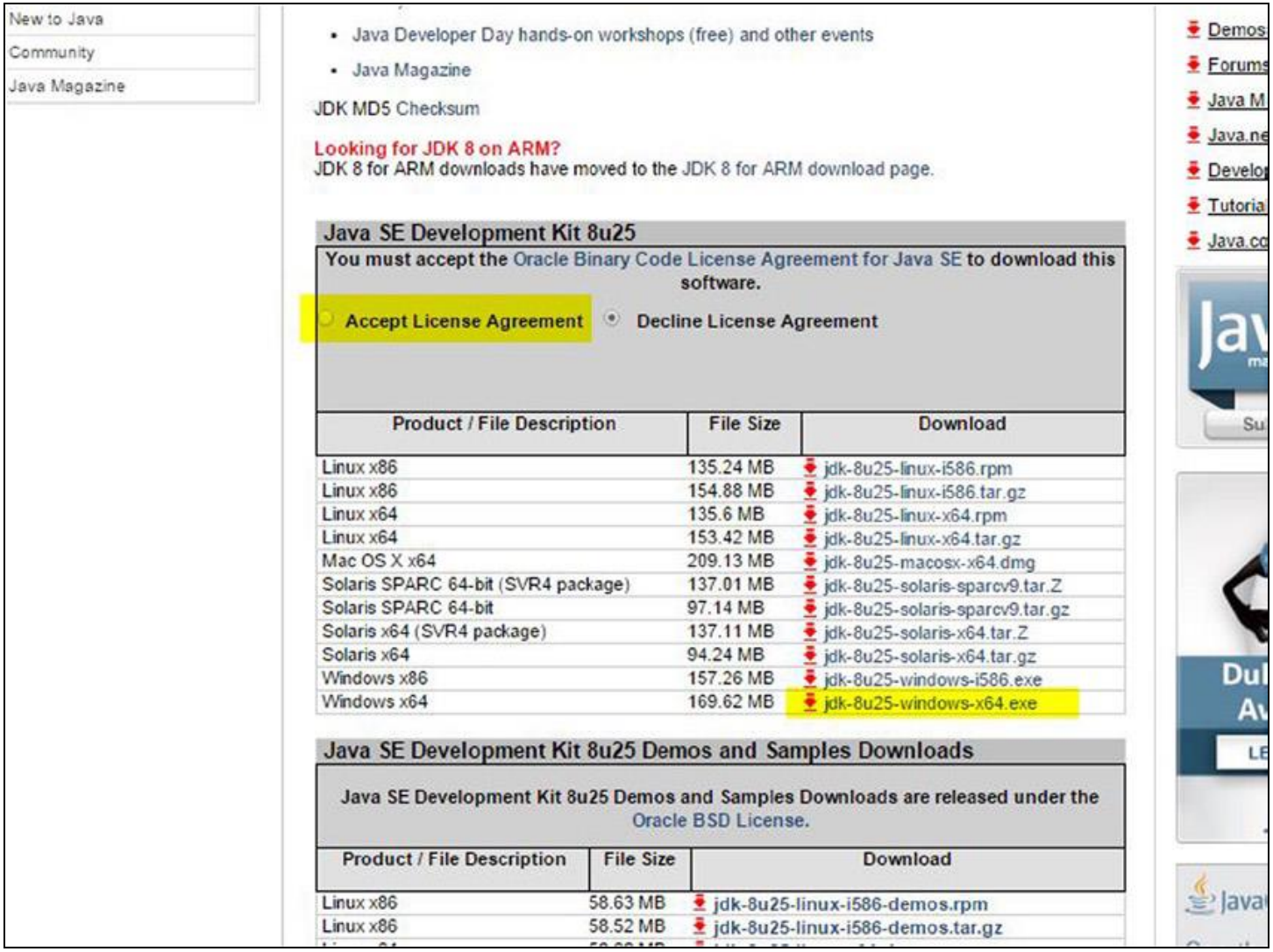
	Windows	OS X	Linux
OS version	Microsoft Windows 10/8.1/8/7/Vista (32 or 64 bit) (64 recommended)	Mac OS X 10.8.5 or higher, up to 10.10 to up 10.10.2 up 10.10.3 on 10.10.5 (Yosemite)	GNOME or KDE or Unity desktop on Ubuntu or Fedora or GNU/Linux Debian
RAM	2 GB RAM minimum, 4 GB RAM recommended, 16 GB RAM good		
Disk space	500 MB disk space		
Space for Android SDK	At least 1 GB for Android SDK, emulator system images, and caches		
JDK version	Java Development Kit (JDK) 7 or higher 64 bit (full JDK not only JRE)		
Processor	I5 orI7 Intel that support Intel® Hardware Accelerated Execution Manager (Intel® HAXM)		
Screen resolution	1280x800 minimum screen resolution		

Downloading and Installing

- JDK (not JRE) must be installed before:

www.oracle.com/technetwork/java/javase/downloads/index.html

on Windows:



New to Java
Community
Java Magazine

- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u25

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement | Decline License Agreement

Product / File Description	File Size	Download
Linux x86	135.24 MB	jdk-8u25-linux-i586.rpm
Linux x86	154.88 MB	jdk-8u25-linux-i586.tar.gz
Linux x64	135.6 MB	jdk-8u25-linux-x64.rpm
Linux x64	153.42 MB	jdk-8u25-linux-x64.tar.gz
Mac OS X x64	209.13 MB	jdk-8u25-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.01 MB	jdk-8u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.14 MB	jdk-8u25-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.11 MB	jdk-8u25-solaris-x64.tar.Z
Solaris x64	94.24 MB	jdk-8u25-solaris-x64.tar.gz
Windows x86	157.26 MB	jdk-8u25-windows-i586.exe
Windows x64	169.62 MB	jdk-8u25-windows-x64.exe

Java SE Development Kit 8u25 Demos and Samples Downloads

Java SE Development Kit 8u25 Demos and Samples Downloads are released under the Oracle BSD License.

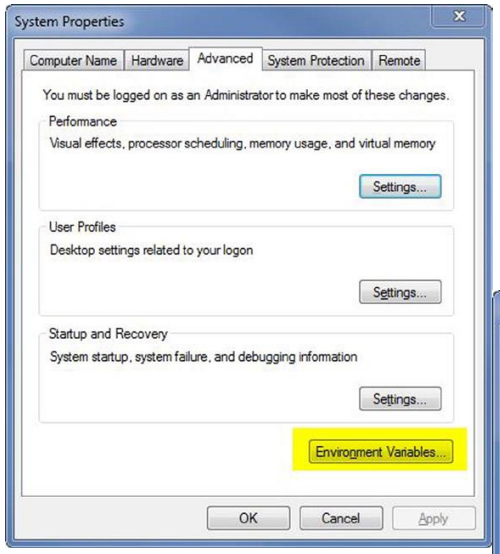
Product / File Description	File Size	Download
Linux x86	58.63 MB	jdk-8u25-linux-i586-demos.rpm
Linux x86	58.52 MB	jdk-8u25-linux-i586-demos.tar.gz

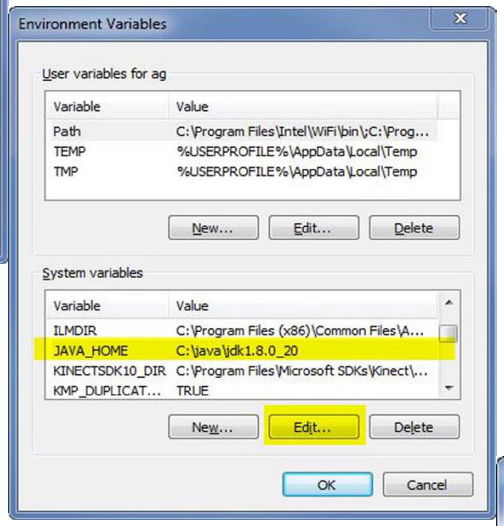
Demos
Forums
Java M
Java.ne
Develop
Tutoria
Java.co

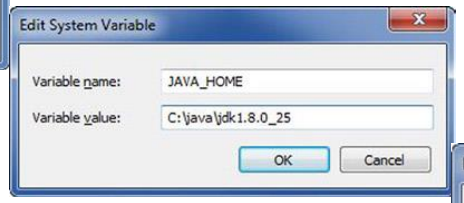
Downloading and Installing

- JDK installation: navigate to the location where installation file was download and execute that file, preferably in C:\Program Files\Java\.

- after this, Environmental Variables on Windows must be configurated:

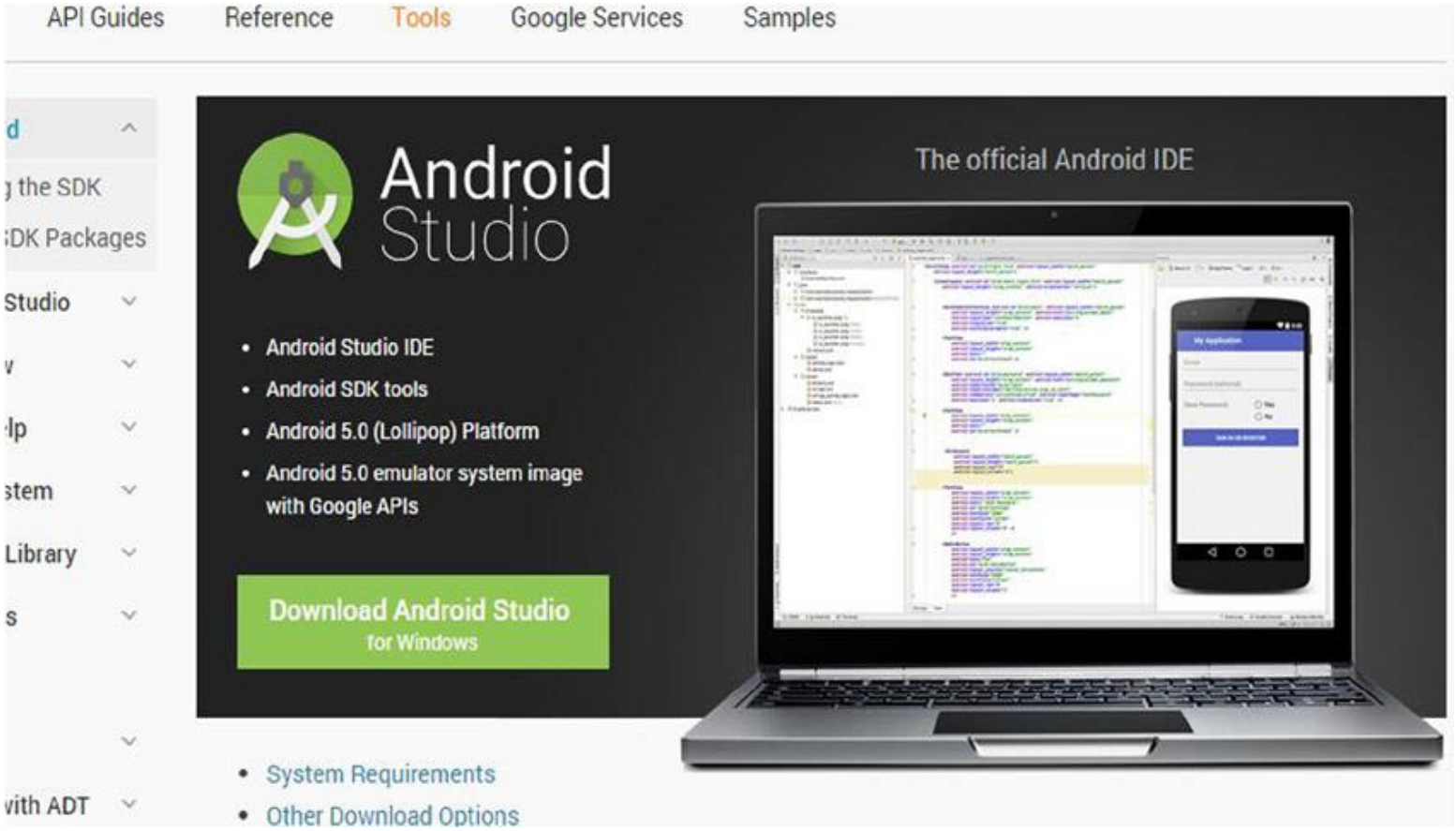






Downloading and Installing

- In general Android Studio is at URL: http://developer.android.com/sdk/index.html
- The package downloaded includes the following: Android Studio bundle of IntelliJ IDEA, Built-in Android SDK, All related Android build tools and Android Virtual Device (AVD)



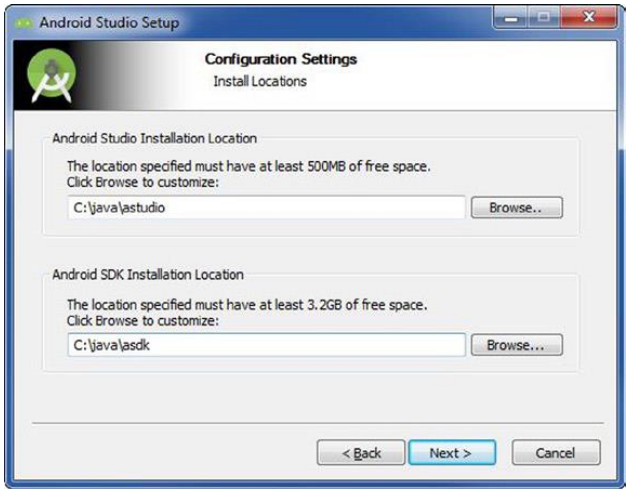
Downloading and Installing

After download:

-for a Windows installation run the executable and follow the prompts to choose an installation path, and all the instalation options;

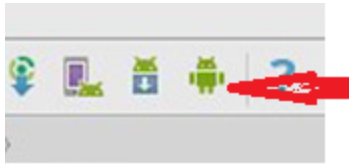
-for OS X, open the .dmg file and copy the Android Studio entry to your Applications folder.

-under Linux, extract the contents of the .tgz file to your desired location.



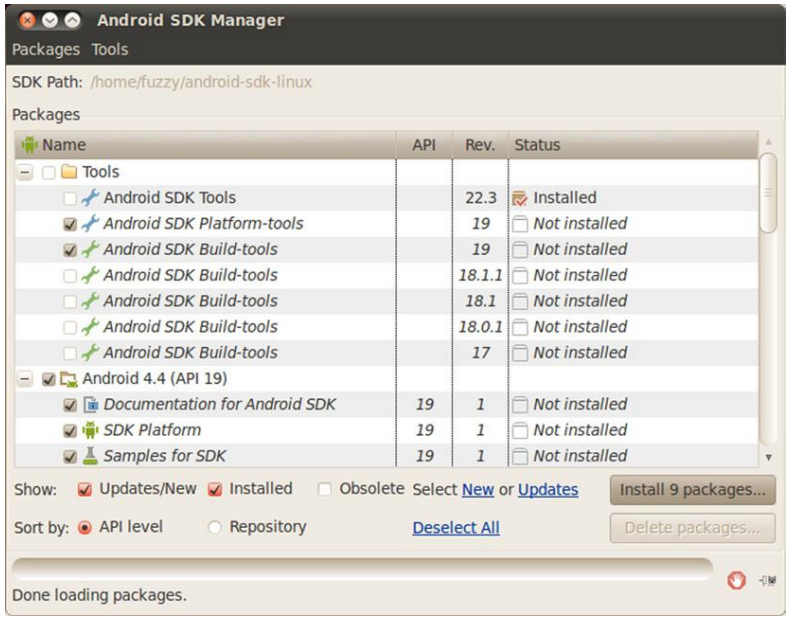
Downloading and Installing

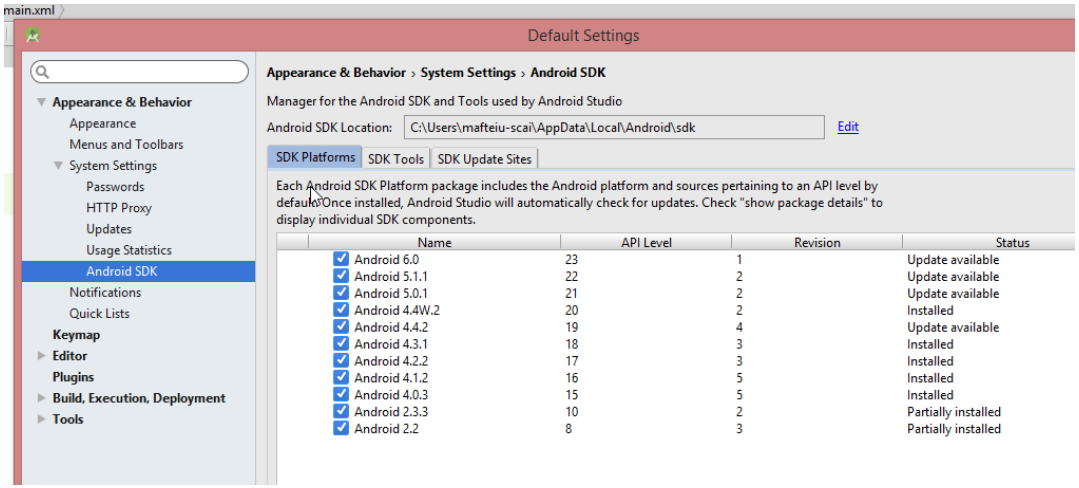
- **Downloading the Android SDK**: to build applications for Android, you need the Android SDK. As stated before, the SDK comes with the base tools. But, you can then, download the package parts that you need and/or want to use. For this you must use *Android SDK Manager.*



Android Studio 1.3

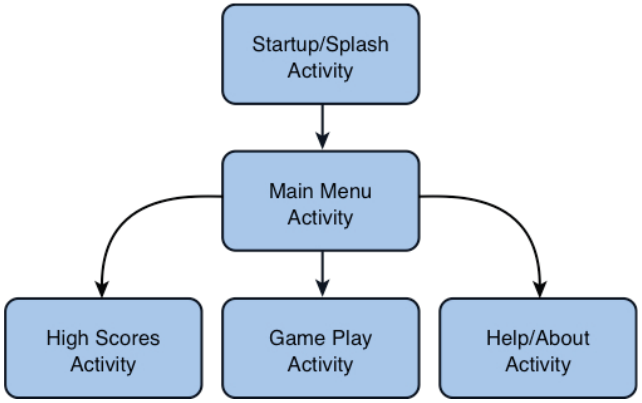
Android Studio 1.4





Android's Fundamental Components

- **View**: user interface (UI) elements (buttons, labels, text fields, etc) that form the basic building blocks of a user interface. *Everything you see is a view.*
- **Activity**: is a UI concept that usually represents a single screen in your application. It generally contains one or more views.
- **Fragment**: in the case of a large screen, it is difficult to manage all of its functionality in a single activity. *Fragments* are like sub-activities, and an activity can display one or more fragments on the screen at the same time.



A simple game with five activities.

- **Intent**: defines an "intention" to do some work. Examples of their use could be: start a service, launch an activity, display a web page, dial a phone number or answer a phone call. Intents are not always initiated by your application, they're also used by the system to notify your application of specific events, such as the arrival of a text message.

Android's Fundamental Components

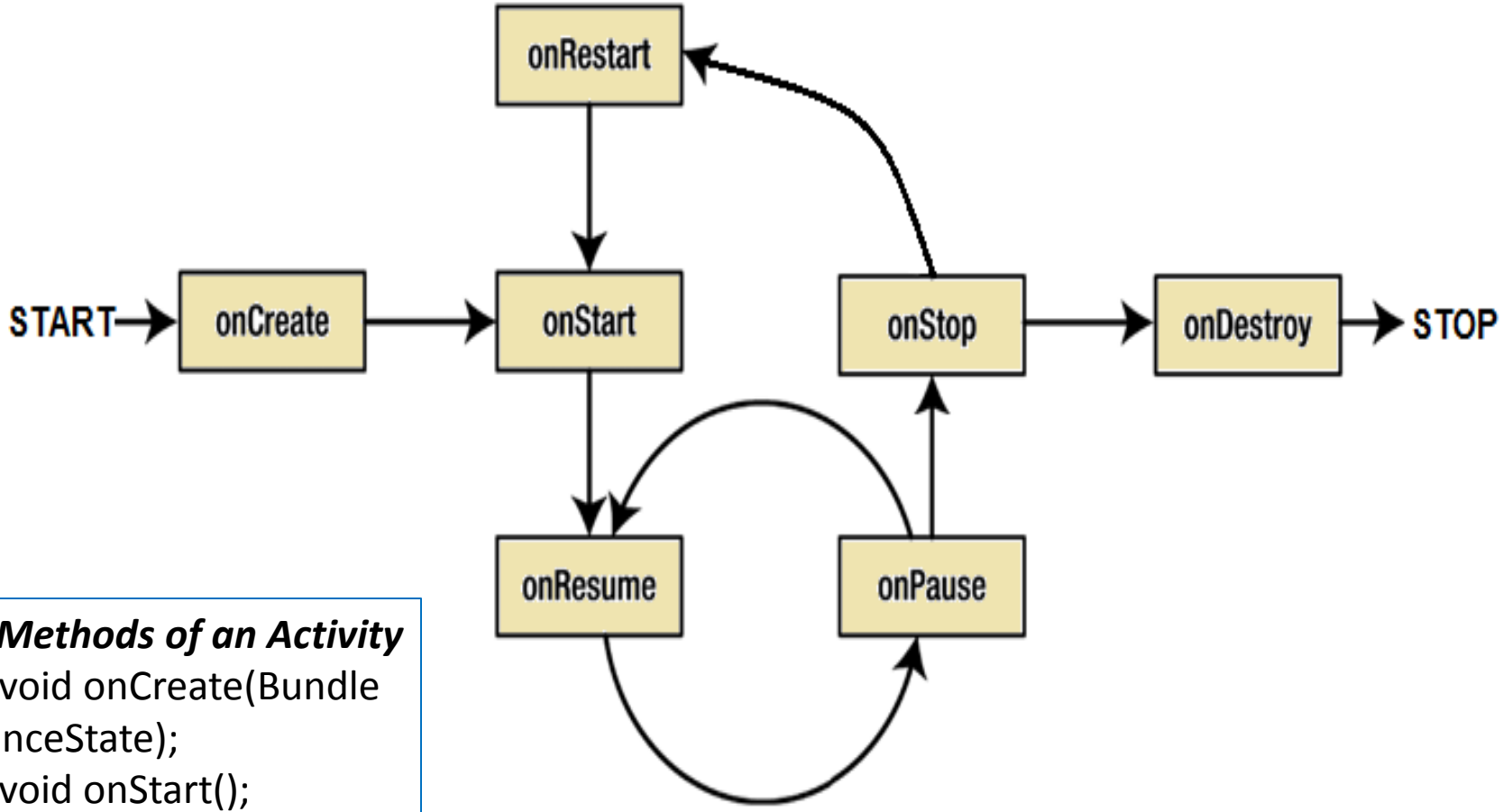
- **Content Provider**: due to data sharing among mobile applications on a device, Android defines a standard mechanism for applications to share data, such as a list of contacts.
- **Service**:

- local services: components that are only accessible by the application that is hosting the service.

- remote services: services that are meant to be accessed remotely by other applications running on the device.

- **AndroidManifest.xml**: defines the contents and behavior of app. For example, it lists app's activities and services, along with the permissions and features the application needs to run.
- **AVD *(Android Virtual Device)***: An AVD represents a device and its configuration. It allows developers to test their applications without hooking up an hardware Android device (smartphone or tablet). With AVD many different types of real devices can be emulated.

Application Life Cycle



State transitions of an Activity

Life-Cycle Methods of an Activity

```
protected void onCreate(Bundle
savedInstanceState);
protected void onStart();
protected void onRestart();
protected void onResume();
protected void onPause();
protected void onStop();
protected void onDestroy();
```

Simulation vs. Emulation

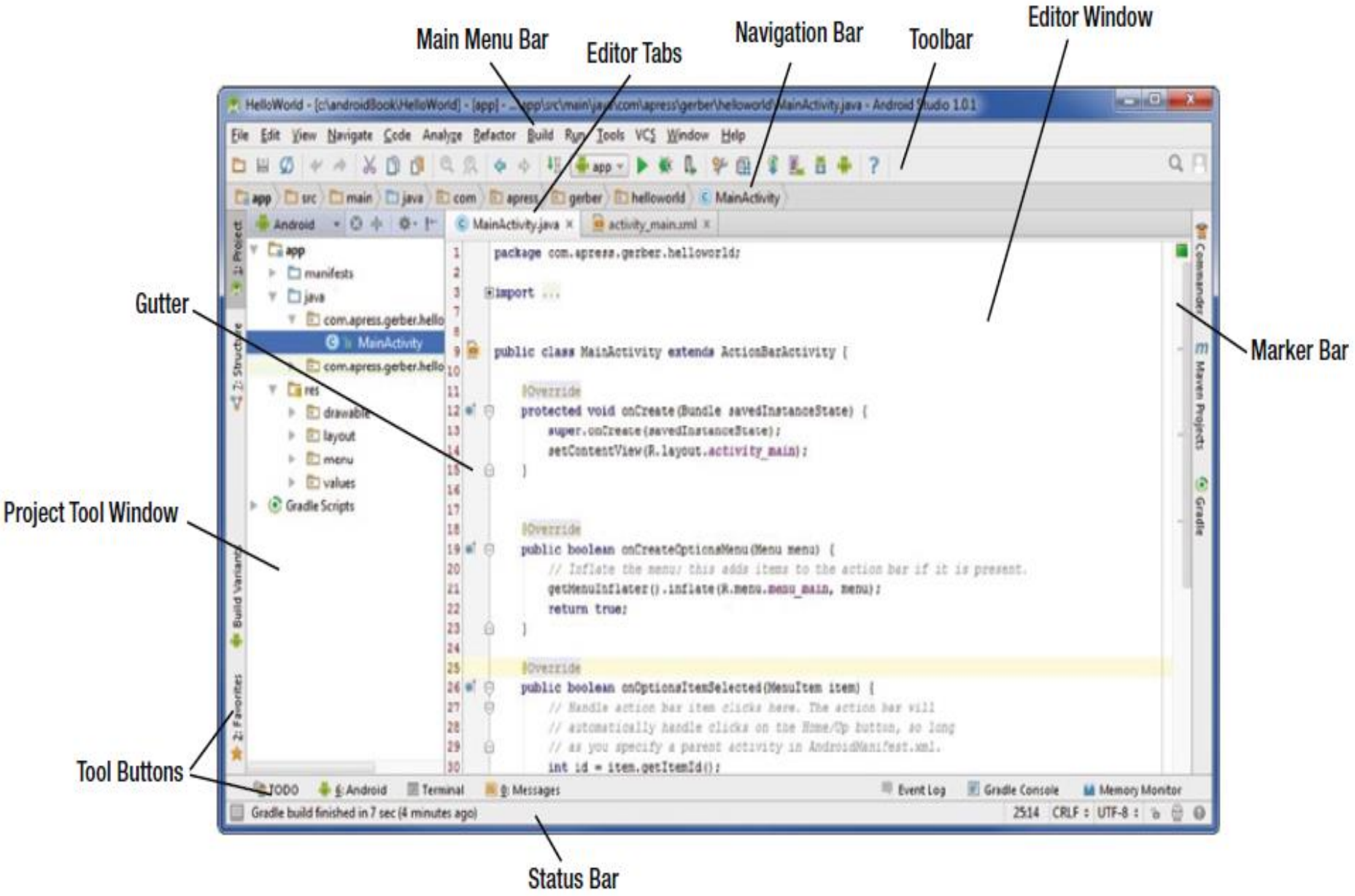
- **Simulation:** the simulated system behaves *similar to* real system. It provides the basic behaviour of a real system, but not all the rules of the real system.
- **Emulation:** the emulated system behaves *exactly like* real system, and respect all the rules of the real system. It is effectively a complete replication of a real system that operate in a different environment

---

- *Simulation: Corona SDK ⇔ Emulation: AVD Android Studio*

- *In Android Studio: AVD Manager is a utility provided by Google which allows to create emulated Android devices. Thus, can be created as many devices are desired, with different hardware specifications and with different screen resolutions.*

Navigating Android Studio



Android Studio's integrated development environment

- Code Folding: hide particular blocks of code

```
3   import android.app.Activity;
4   import android.support.v7.app.AppCompatActivity;
5   import android.os.Bundle;
6   import android.view.Menu;
7   import android.view.MenuItem;
8   import android.widget.EditText;
9   import android.widget.Button;
10  import android.view.View;
11
12  public class MainActivity extends Activity
```

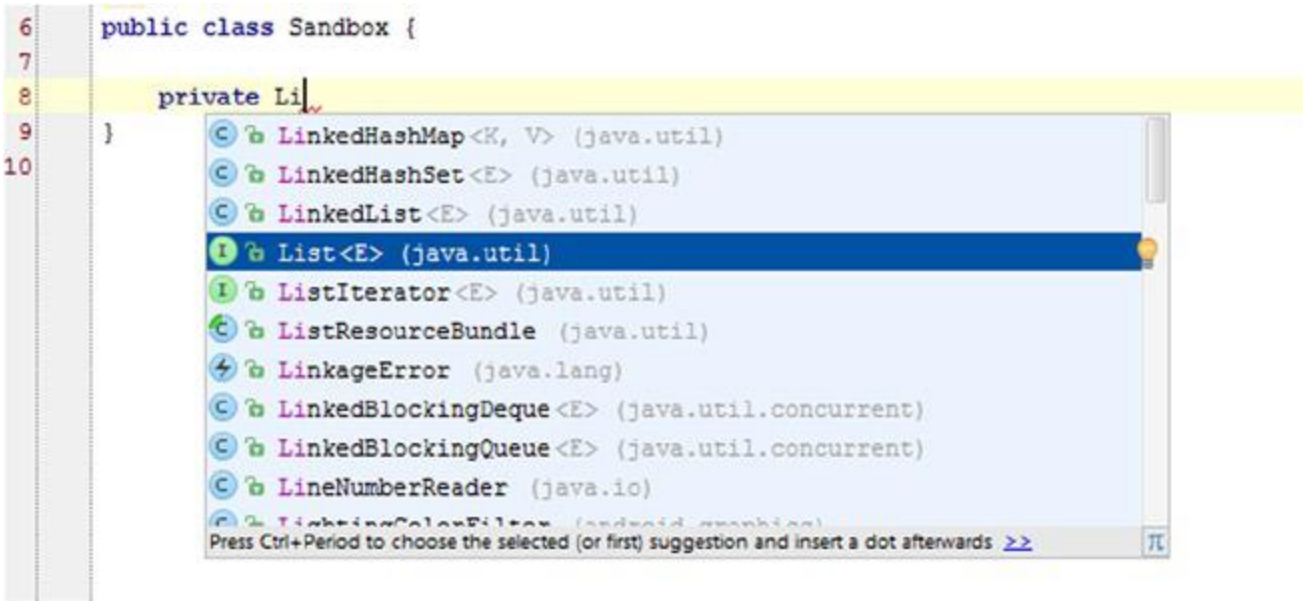
```
31
32      private void setupButtons() {...}
42
43      @Override
44      public void onClick(View v) {...}
65
66      private double plus(double n1, double n2) { return n1 + n2; }
```

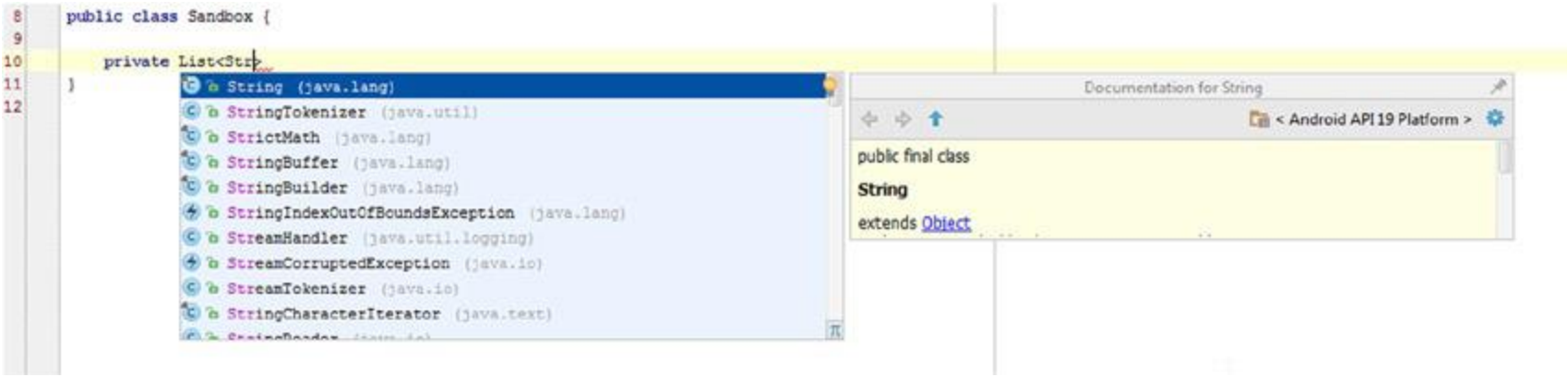
folding outline

```
2
3   import ...
11
```

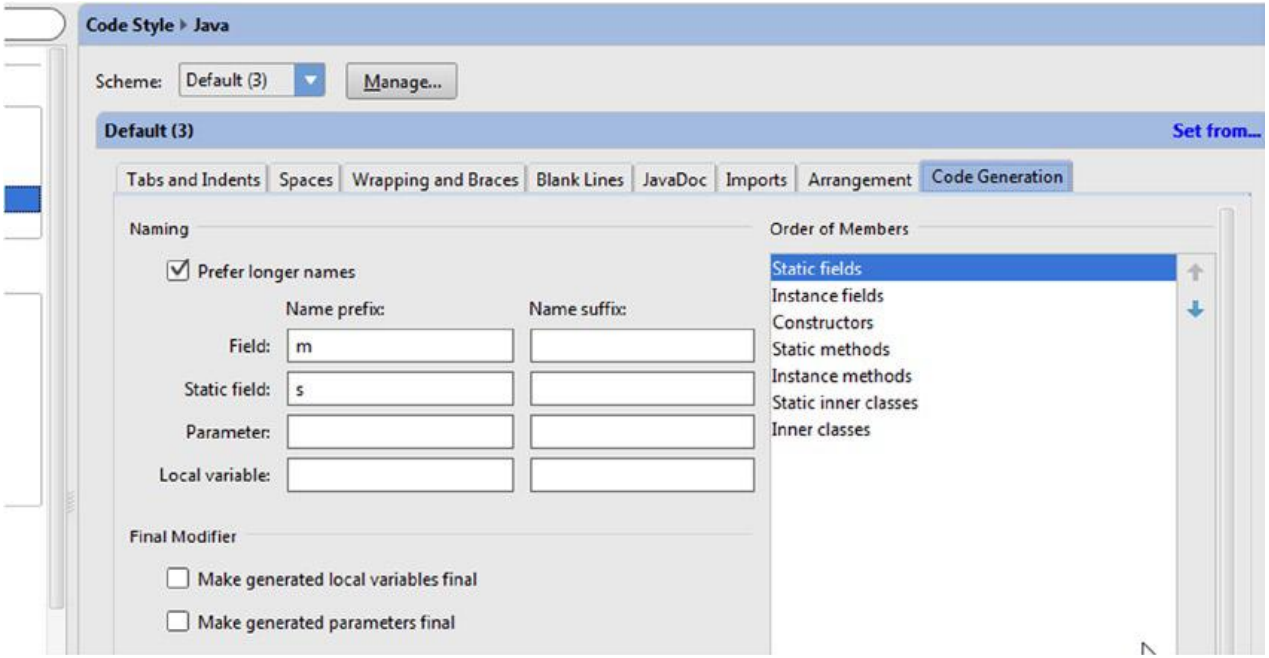
```
31
32      private void setupButtons() {
33          Button b = (Button) this.findViewById(R.id.plusButton);
34          b.setOnClickListener(this);
35          b = (Button) this.findViewById(R.id.minusButton);
36          b.setOnClickListener(this);
37          b = (Button) this.findViewById(R.id.multiplyButton);
38          b.setOnClickListener(this);
39          b = (Button) this.findViewById(R.id.divideButton);
40          b.setOnClickListener(this);
41      }
42
43      @Override
44      public void onClick(View v) {...}
65
```

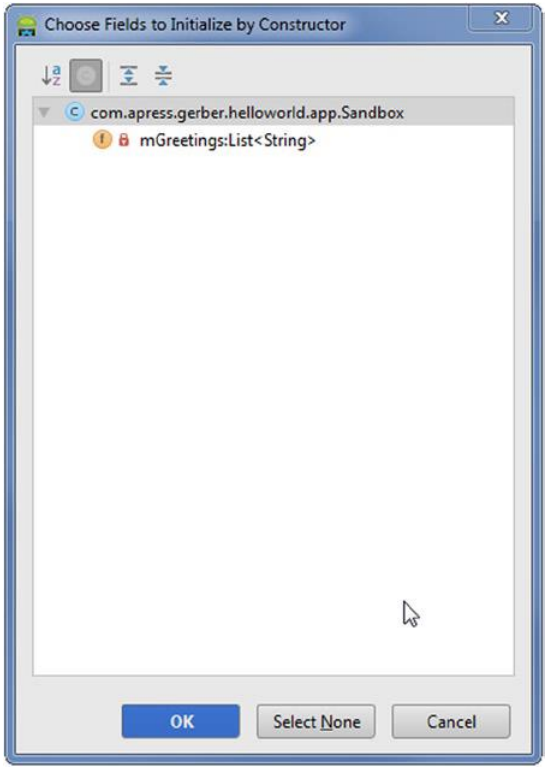
- Code Completion





- **Code Generation:** generate methods (constructors, getters, setters, equals(), hashCode(), toString(), and so on).





Styling The Code:
Auto-Indent Lines
Rearrange Code
....

Ta-Ta for now!