SMARTPHONE APPLICATION DEVELOPMENT WITH GOOGLE'S ANDROID

School of Engineering and Computer Science
Fairfield University
Fall 2012
November 29, 2012
By

FAHIM UDDIN
http://fahim.cooperativecorner.com
email@fahim.cooperativecorner.com
203-543-9688

INTRODUCTORY HANDOUTS

From the Contents, The topics (1,2,3,4,10,11) will be discussed during lecture in classroom. Topics (5,6,7,8,9) are both given in handouts and will be discussed in class as well.

This document is very basic introduction to Android. Sole purpose is to let you read some basics and make you get your machines ready with necessary tools and discuss very first program to start getting our hands wet with it.

As you will see, Android is much-much-much-much more than that.

These handouts are written with sense of humor in mind so you don't fall asleep. Feel free to point out any technical error in here and I will reward you with $1 Bill with my signature on it. ☺ - believe me?

Contents

7. Preparing your machines (Handout)
    i) Hardware requirements
    ii) Software requirements
8. Hello Android Example with sample code. (Handout and in Class)
9. Study resources. (Handout)
10. Conclusion. (In Class)
11. Next Lecture coverage. (In Class)

5. INTRODUCTION TO ANDROID

It was originally developed by Android Inc. Google purchased it in 2005. Today it is being developed by OHA(Open Handset Alliance) led by Google and other companies.

OHA – a consortium of about 86 hardware, software and telecommunication companies who are devoted to promoting and advancing Open Standard for mobile devices. Google has releases Android code as Open Source under the Apache License. This License simply means: ***go ahead and download the source code (SDK) and develop it for free***. Vendors and hardware manufacturer can extend android into their own versions to distinguish their products and devices.

Let me have you pick the definition you like.

a) Android is simply a mobile operating system. It is actually modified version of Linux OS.
b) The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation. (http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html)
c) It is an open source mobile operating system that combines and builds on different parts of various open source projects. This makes it platform independent development tool.
d) It is simply another OOP framework that helps building mobile devices environment that is running supported OS in conjunction. For example, the application that you develop using Android can run on any phone or tablet device that is running any of Google's OS versions (will discuss this later) and it does not matter who is manufacturer of device, unlike Apple App Development that requires Apple Device with iOS running as Operating System.
e) If you come up with your own definition of Android after your hands are wet enough with experience of development with Android, then please write that in lines below:

YOUR DEFNITION:____

____

____

____

6. INTRODUCTION TO JAVA AND ECLIPSE

For the purpose of this introductory lecture series, I will not dig into Java. You are encouraged to take Java courses otherwise if you are interested in Java as a Language and or a separate credit course. However, I highly recommend you to go through online documentation (http://docs.oracle.com/javase/tutorial/java/concepts/). For regular 3 credit course class of Android Application Development, I will spend a lecture or two on Java to build basics upon which we can learn Android development.

You should also have received a PDF file along with these handouts for Java. Go through if you are brand new to Java.

Eclipse:

This is one of freely available and highly used and recommended IDE for developing Android (Java based) applications.

Take a look at the following recourses for more details.

1. http://www.eclipse.org/
2. http://www.vogella.com/articles/EclipseWizards/article.html
3. http://www.eclipse.org/resources/

7. PREPARING YOUR MACHINES

Hardware requirements:

Source:
http://www.netmite.com/android/mydroid/development/pdk/docs/system_requirements.html

Device Requirements

While Android is designed to support a wide variety of hardware platforms and configurations, this section provides recommended minimum device requirements.

Feature	Minimum Requirement	Notes
Chipset	ARM-based	For the first release, Android is primarily targeted towards mobile handsets and portions of the platform, such as Dalvik VM graphics processing, currently assume an ARM architecture.
Memory	128 MB RAM; 256 MB Flash	Android can boot and run in configurations with less

	External	memory, but it isn't recommended.
Storage	Mini or Micro SD	Not necessary for basic bring up, but recommended.
Primary Display	QVGA TFT LCD or larger, 16-bit color or better	The current Android interface targets a touch-based HVGA resolution display with a touch-interface no smaller than 2.8 inches in size. However, smaller displays will suffice for initial porting.
Navigation Keys	5-way navigation with 5 application keys, power, camera and volume controls	
Camera	2MP CMOS	Not required for basic bring up.
USB	Standard mini-B USB interface	Android uses the USB interface for flashing the device system images and debugging a running device.
Bluetooth	1.2 or 2.0	Not required for initial bring up.

If available, your Android device can also benefit from the following optional device characteristics:

- QWERTY keyboard
- WiFi
- GPS

Software Requirements

Supported OS:

•Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)

•Mac OS X 10.5.8 or later (x86 only)

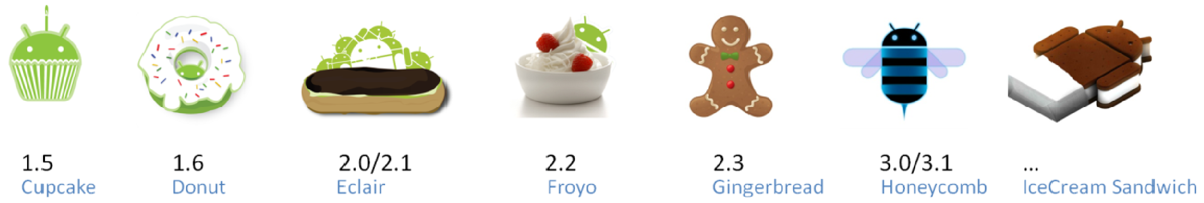
•Linux (tested on Ubuntu Linux, Lucid Lynx)

◦GNU C Library (glibc) 2.7 or later is required.

◦On Ubuntu Linux, version 8.04 or later is required.

◦64-bit distributions must be capable of running 32-bit applications.

<u>Various version of Android OS:</u>



Android 2.2 (Called Froyo. The oldest of the android operating system).
Android 2.3 (operating system. called Gingerbread. Supports Adobe flash).
Android 3.0 (only for tablets, usually 7 to 10 inches. Called Honeycomb)
Android 4.0 (only on Nexus cell phone and the newest tablets. Called Ice cream)

* For cell phones, I would recommend getting Android 2.3. Make sure that it is running Android 2.3 or Gingerbread. I would say 90 percent of the cell phones already have Gingerbread running on the android cell phones.

Setup Checklist

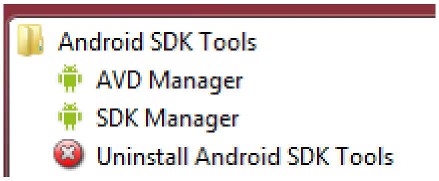
Source : http://developer.android.com/

- Step1: Download your eclipse IDE (Galileo) or use your existing IDE.
- Step2: Follow the android link section "Installing the ADT Plugin" for eclipse.
- Step3: Download the SDK android-sdk_r04-windows.zip and save it in a location.
- Step4: Unzip the SDK and click on the android-sdk-windows\SDK Setup.exe
- Step5: You will see a window popup, for some reason it fails saying HTTPS.
- Step6: Go to settings check the checkbox "force hhtps://.....".
- Step7: Now you should see list of Android SDK's,
- Step8: Choose , 1. SDK platform Android 2.0.1, API 6, Revision 1, 2. Android SDK Tools, revision 4,3.Google APIs by Google Inc., Android API 6, revision 1, 4.Documentation if you want.

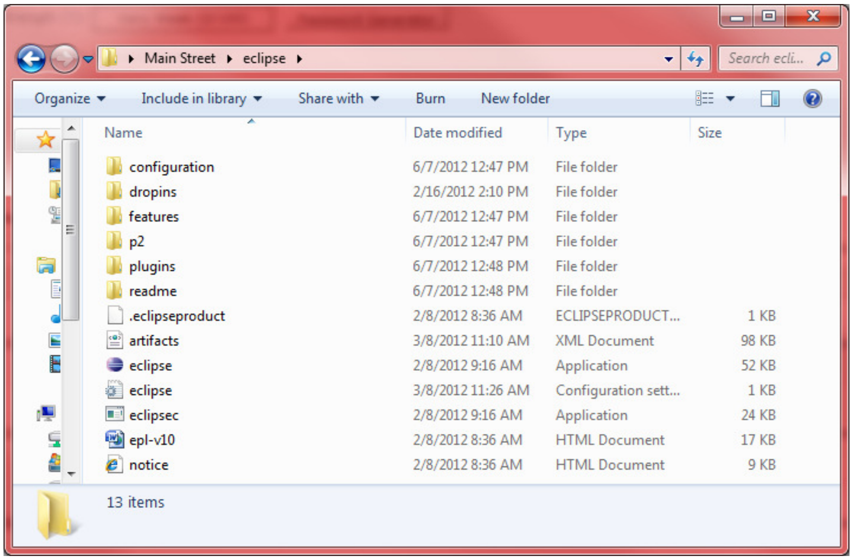
- Step9: Accept all, will install the sdk in android-sdk-windows\platforms\android-2.0.1
- Step10: Go back to eclipse windows -> preferences -> Android -> set your sdk unziped location and save it.

8. HELLO ANDROID EXAMPLE WITH SAMPLE CODE

Before you begin, make sure that you see the following in your program files menu if installed with default settings.



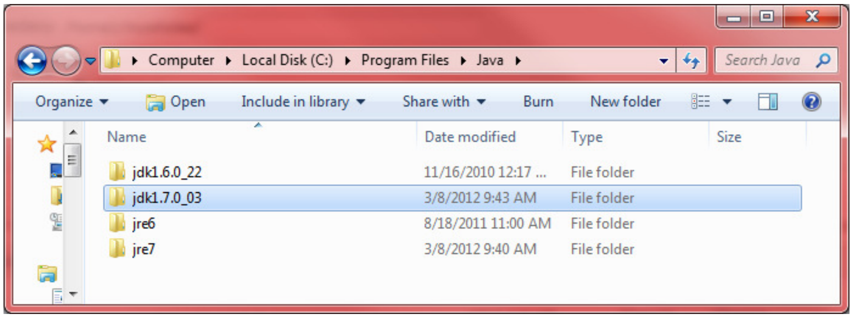
Eclipse IDE is installed somewhere on your file system on your workstation.



Name	Date modified	Type	Size
configuration	6/7/2012 12:47 PM	File folder	
dropins	2/16/2012 2:10 PM	File folder	
features	6/7/2012 12:47 PM	File folder	
p2	6/7/2012 12:47 PM	File folder	
plugins	6/7/2012 12:48 PM	File folder	
readme	6/7/2012 12:48 PM	File folder	
.eclipseproduct	2/8/2012 8:36 AM	ECLIPSEPRODUCT...	1 KB
artifacts	3/8/2012 11:10 AM	XML Document	98 KB
eclipse	2/8/2012 9:16 AM	Application	52 KB
eclipse	3/8/2012 11:26 AM	Configuration sett...	1 KB
eclipsec	2/8/2012 9:16 AM	Application	24 KB
epl-v10	2/8/2012 8:36 AM	HTML Document	17 KB
notice	2/8/2012 8:36 AM	HTML Document	9 KB

13 items

Don’t forget Java JDK



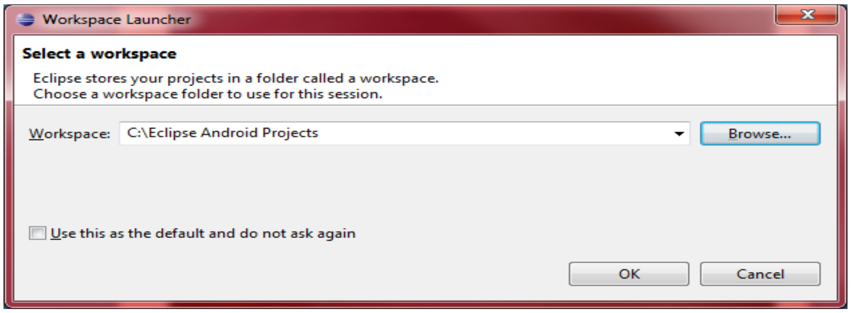
Name	Date modified	Type	Size
jdk1.6.0_22	11/16/2010 12:17 ...	File folder	
jdk1.7.0_03	3/8/2012 9:43 AM	File folder	
jre6	8/18/2011 11:00 AM	File folder	
jre7	3/8/2012 9:40 AM	File folder	

Alright. That is good for now.

Let open the Eclipse,

Double click the eclipse Icon that has type as Application:

After splash screen disappears, you would see a dialog box where you need to provide IDE with a location of a folder where you like your projects to be saved as file system. I would create a folder in C drive as the following:



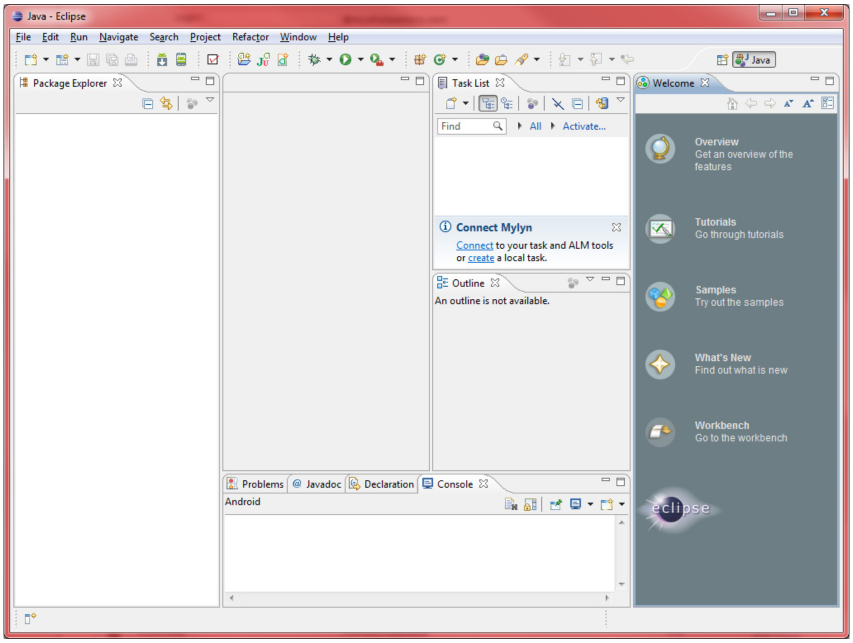
Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\Eclipse Android Projects

☐ Use this as the default and do not ask again

Hang in there guys after clicking on OK button, you would soon see the screen as below screen shot. Yes, I know it took little while (could be slow).



Connect Mylyn

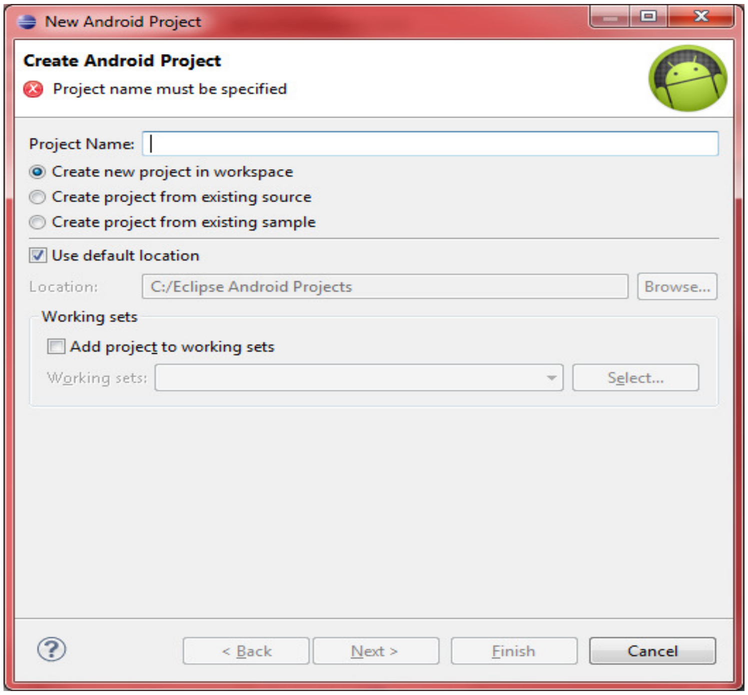
Connect to your task and ALM tools or create a local task.

Outline

An outline is not available.

If you like to dig more into IDE in general, then play with it as much as possible, I will skip to the our business directly and that is **creating brand new Android project.**

Go to File Menu and follow New to Android Project type and you would get as screen below.



Create Android Project

☒ Project name must be specified

Project Name: |

● Create new project in workspace

○ Create project from existing source

○ Create project from existing sample

☑ Use default location

Location: C:/Eclipse Android Projects

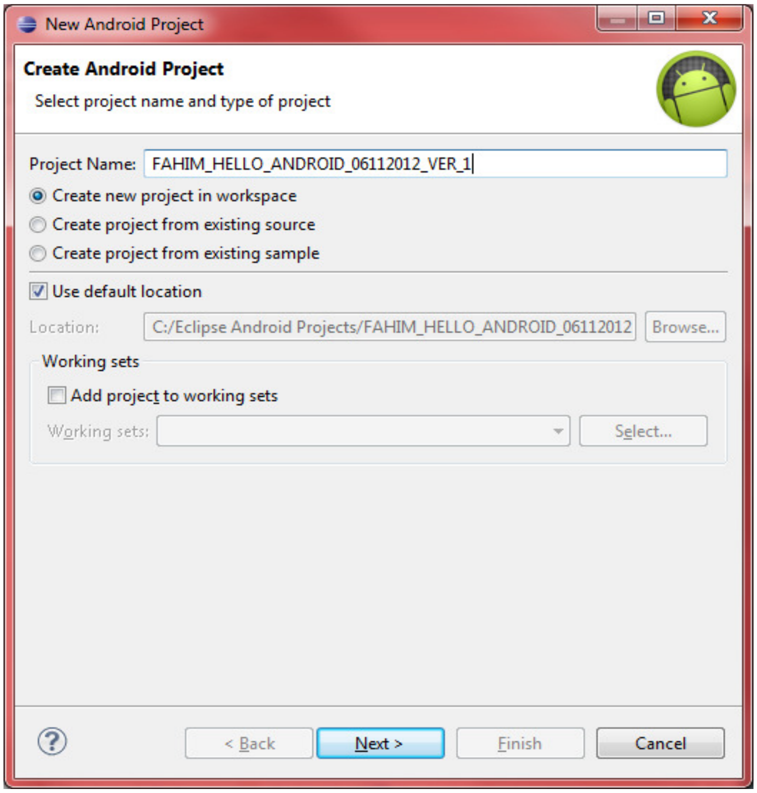
Working sets

☐ Add project to working sets

Working sets:

Follow the following format for any project name. My name is FAHIM. You can have your own name if you don't like mine.

FAHIM_ProjectName_ANDROID_MMDDYYYY_VER_1



New Android Project

Create Android Project

Select project name and type of project

Project Name: FAHIM_HELLO_ANDROID_06112012_VER_1

● Create new project in workspace

○ Create project from existing source

○ Create project from existing sample

☑ Use default location

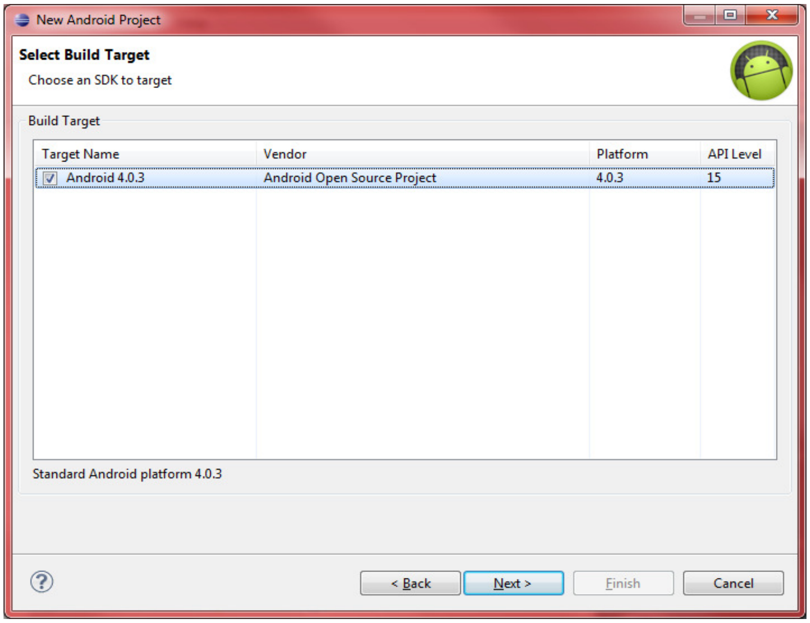
Location: C:/Eclipse Android Projects/FAHIM_HELLO_ANDROID_06112012

Working sets

□ Add project to working sets

Working sets:

Do you really want me to tell you what to do here?



New Android Project

Select Build Target

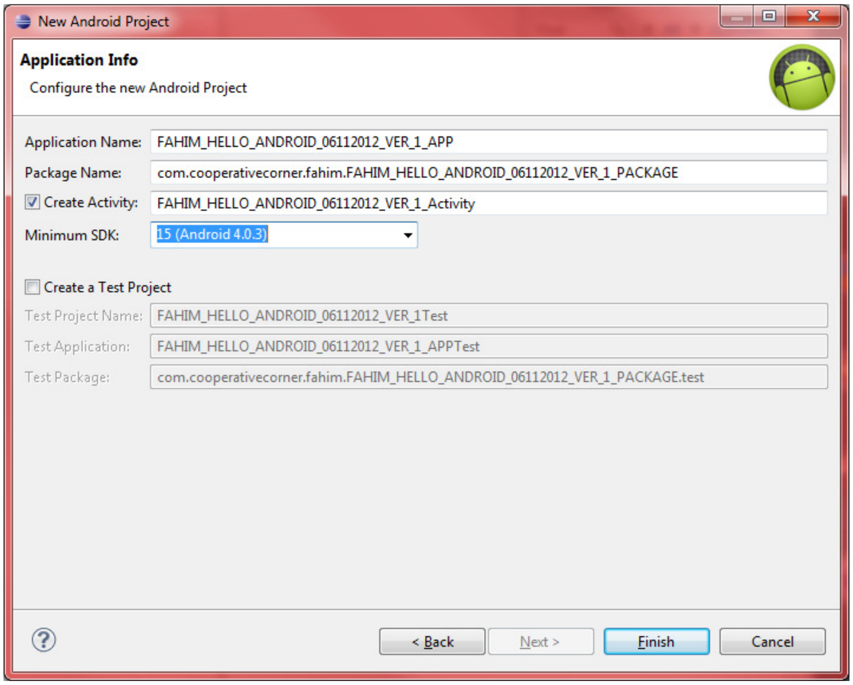
Choose an SDK to target

Build Target

Target Name	Vendor	Platform	API Level
☑ Android 4.0.3	Android Open Source Project	4.0.3	15

Standard Android platform 4.0.3

I will discuss various choices during lecture on this window.



Application Info

Configure the new Android Project

Application Name: FAHIM_HELLO_ANDROID_06112012_VER_1_APP

Package Name: com.cooperativecorner.fahim.FAHIM_HELLO_ANDROID_06112012_VER_1_PACKAGE

☑ Create Activity: FAHIM_HELLO_ANDROID_06112012_VER_1_Activity

Minimum SDK: 15 (Android 4.0.3)

□ Create a Test Project

Test Project Name: FAHIM_HELLO_ANDROID_06112012_VER_1Test

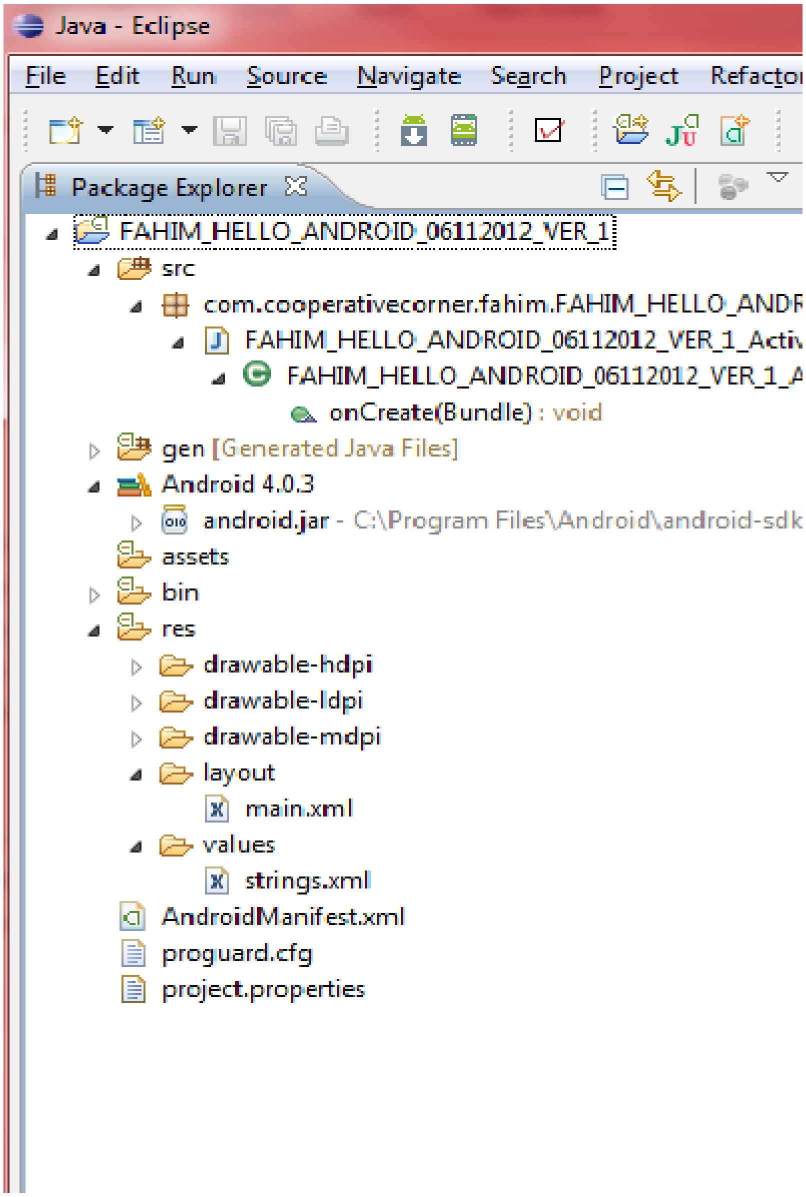
Test Application: FAHIM_HELLO_ANDROID_06112012_VER_1_APPTest

Test Package: com.cooperativecorner.fahim.FAHIM_HELLO_ANDROID_06112012_VER_1_PACKAGE.test

When you click finish, You are done. Good night.

☺

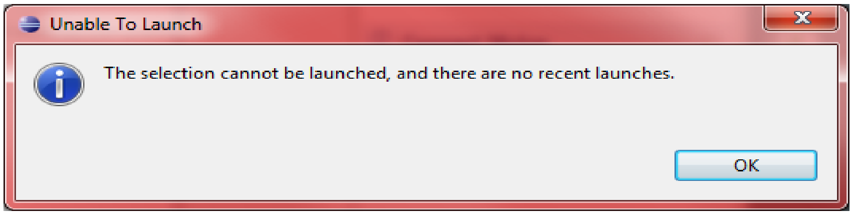
Eclipse will provide you with basic structure of your application and folders with sample code already typed in associated files. Take a look at image on next page.



```
FAHIM_HELLO_ANDROID_06112012_VER_1
  src
    com.cooperativecorner.fahim.FAHIM_HELLO_ANDR
      FAHIM_HELLO_ANDROID_06112012_VER_1_Activ
        FAHIM_HELLO_ANDROID_06112012_VER_1_A
          onCreate(Bundle) : void
  gen [Generated Java Files]
  Android 4.0.3
    android.jar - C:\Program Files\Android\android-sdk
  assets
  bin
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  proguard.cfg
  project.properties
```

Let’s run it as is.

Click on Start Green Icon



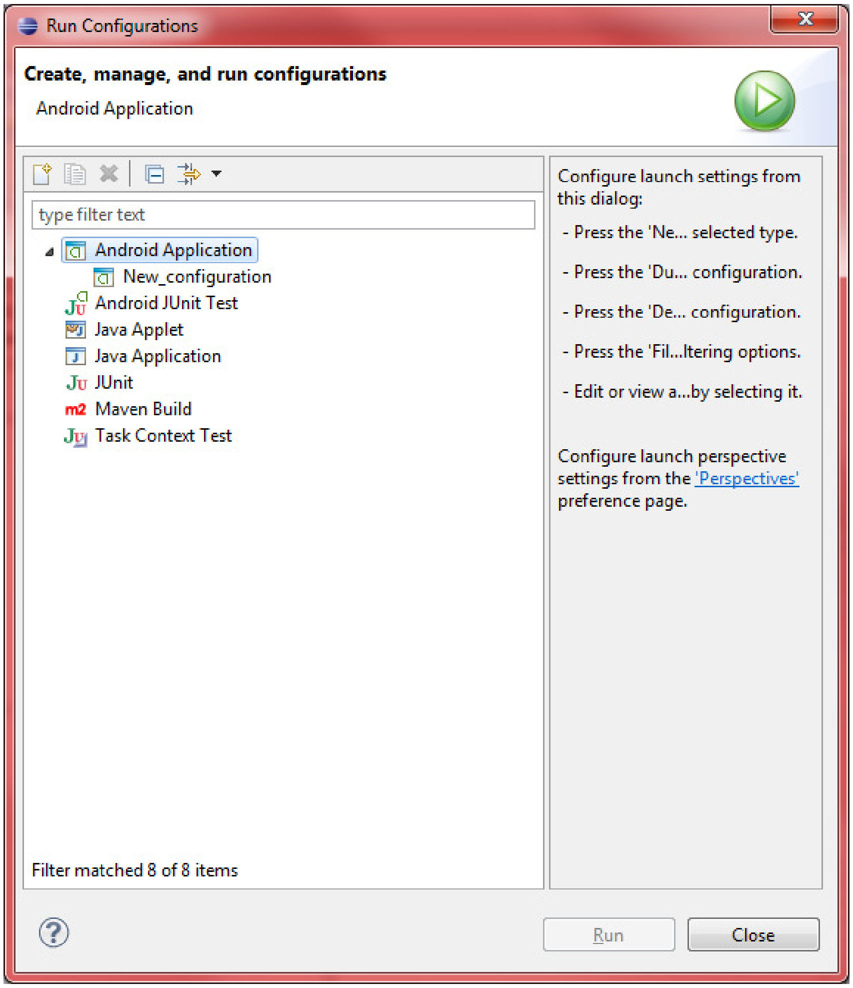
The selection cannot be launched, and there are no recent launches.

Hmm. Why is it?

Let’s take care of this issue:

Go and Run Configurations:

(from where? I let you figure this out)



Run Configurations

Create, manage, and run configurations

Android Application

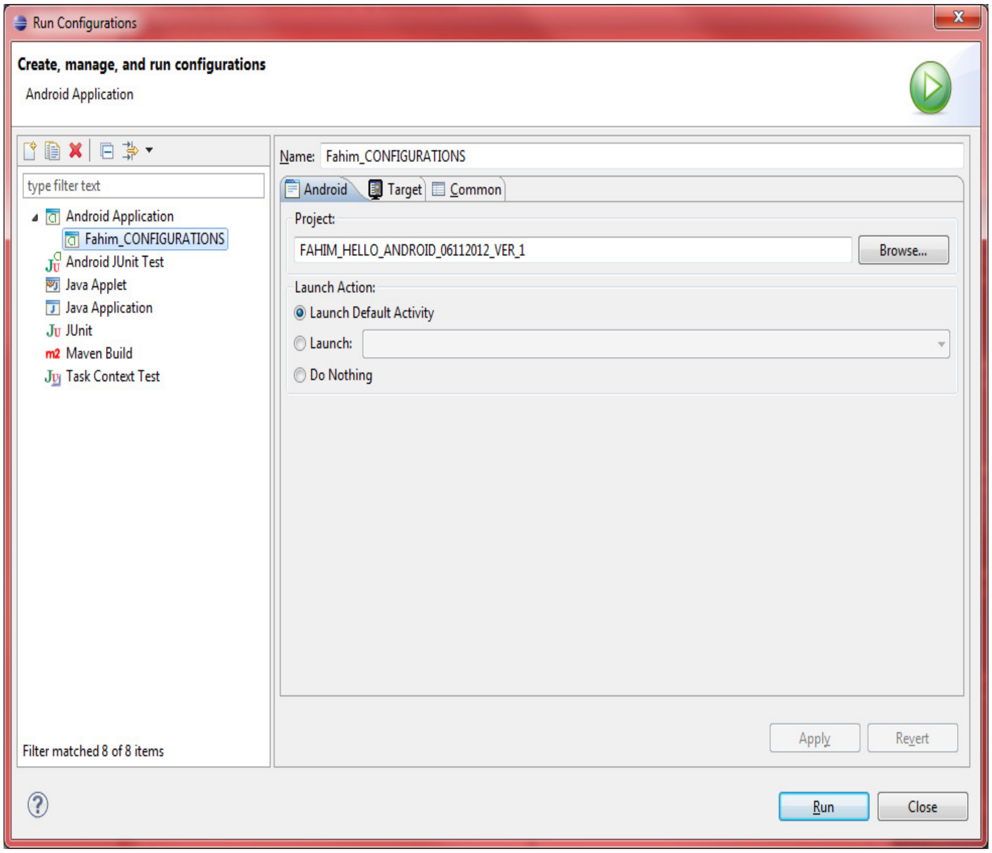
Configure launch settings from this dialog:

- Press the 'Ne... selected type.
- Press the 'Du... configuration.
- Press the 'De... configuration.
- Press the 'Fil...ltering options.
- Edit or view a...by selecting it.

Configure launch perspective settings from the 'Perspectives' preference page.

Filter matched 8 of 8 items

I will discuss Virtual Device and Android Simulator in Class.



Create, manage, and run configurations

Android Application

Name: Fahim_CONFIGURATIONS

Project:

FAHIM_HELLO_ANDROID_06112012_VER_1

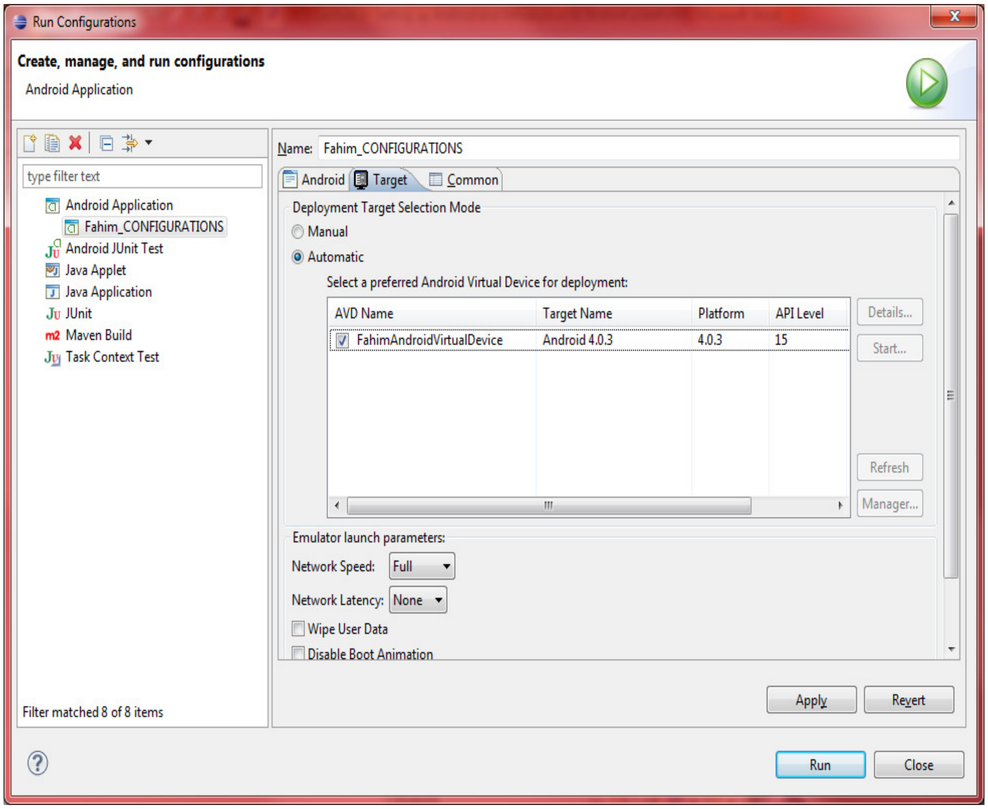
Launch Action:

● Launch Default Activity

○ Launch:

○ Do Nothing

Filter matched 8 of 8 items



Create, manage, and run configurations

Android Application

Name: Fahim_CONFIGURATIONS

Deployment Target Selection Mode

○ Manual

● Automatic

Select a preferred Android Virtual Device for deployment:

AVD Name	Target Name	Platform	API Level
☑ FahimAndroidVirtualDevice	Android 4.0.3	4.0.3	15

Emulator launch parameters:

Network Speed: Full

Network Latency: None

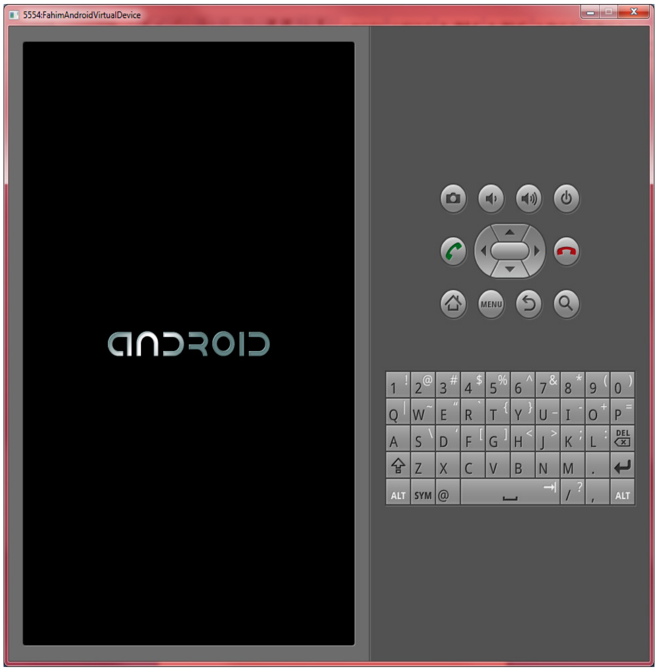
□ Wipe User Data

□ Disable Boot Animation

Filter matched 8 of 8 items

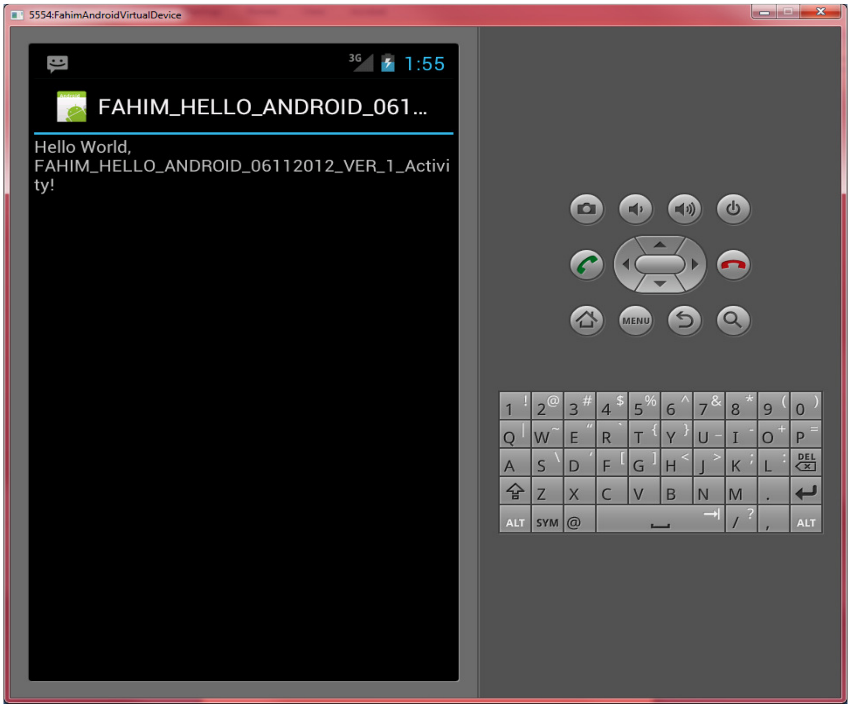
Click on RUN.

If you see this screen, then you don’t have any compilation errors.



This will take few seconds for your program to load up finally so hang in there, but meanwhile keep looking at it, since **Android does not like be ignored.**

Unlock the screen, when home screen appears, and you should see, because I say so. ☺



FAHIM_HELLO_ANDROID_061...

```
Hello World,
FAHIM_HELLO_ANDROID_06112012_VER_1_Activi
ty!
```

Folder Structure Review/ Project Tree

Grandfather node here is Project Node:

FAHIM_HELLO_ANDROID_06112012_VER_1

You have the following sub nodes or folders

- src
- gen
- assets
- bin
- res
- res → layout
- res → values

Rest of the folders are not important to talk about at this point.

What are the files that we have deal with?

FILE 1: Developer baby for Java code, ___Activity.java

This file contains basic java code with onCreate method created by your system. All of your custom java code goes here.

FILE 2: main.xml (under res → layout)

When you open it, you would see two tabs. Graphic layout and xml code.

Switch to xml code for now.

Here you see xml code that basically defines the layout of what you see on screen. This is developer second coding baby.

FILE 3: strings.xml (under res → values)

Similar to main.xml, This is Developer's third coding baby.

FILE 4: R.java (under gen → package name)

This is **not** your 4th baby. **Stay away from it. And don't make any changes in it**. You don't have legal authority on it and I am afraid, you can never adopt this child. Yes, you can take a look and through your other babies, you can see the impact on this reserved child of the system. So in other words, it is a friend of your babies, but not your baby at all.

Yes, it is java file. But not for you to code in but for system to update code. Let system worry about it. **You have enough to worry about elsewhere in project.**

LET'S DO SOME WORK GUYS of YOUR OWN, NOW!

We will make some changes in files (your babies) and then run it to see reflected changes in our UI. Next, we will then talk about each file purpose and how each work together to make a happy family of your first program. – (hey you back there, focus here. No texting in class ☺).

➔ Just Edit code (I will not give you technical explanation at this section, but later).

I am going to add a button to my UI on screen, with caption "My First Button".

Add the following code to R.java???

Nooooooooooooooooooooooooooooooooooooooooooooooooooooo.

Neverrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr.

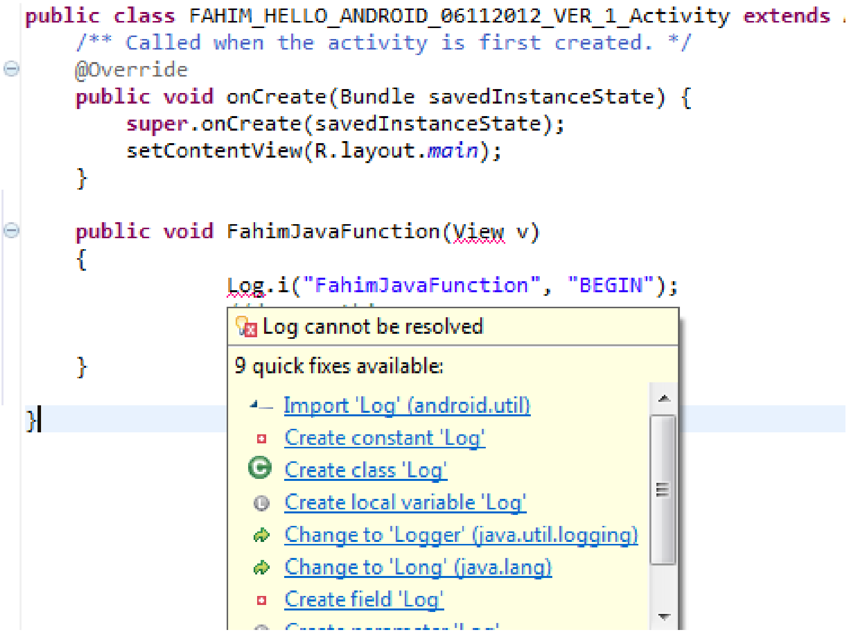
Rather add this to our layout file, which is main.xml

```
<Button

    android:layout_width = "fill_parent"

    android:layout_height = "wrap_content"

    android:text=" My First Button"

    android:onClick="FahimJavaFunction"

/>
```

Add the following code to activity.java file

```
 public void FahimJavaFunction(View v)
 {
                Log.i("FahimJavaFunction", "BEGIN");
                //do something
                Log.i("FahimJavaFunction", "End");
 }
```

You would notice a small red sign that it does not know, what is “Log”, using mouse intellisense to add the required package: and it will add



```
public class FAHIM_HELLO_ANDROID_06112012_VER_1_Activity extends
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void FahimJavaFunction(View v)
    {
                Log.i("FahimJavaFunction", "BEGIN");


    }

}
```

Log cannot be resolved

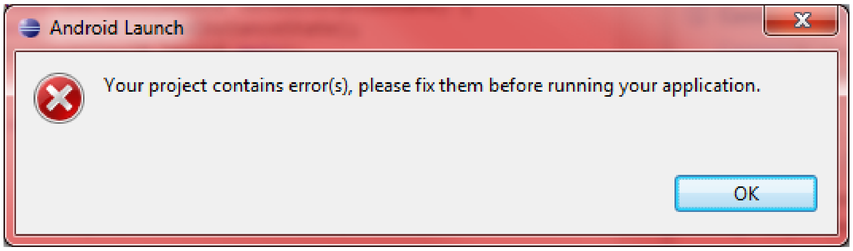
9 quick fixes available:

Import 'Log' (android.util)
Create constant 'Log'
Create class 'Log'
Create local variable 'Log'
Change to 'Logger' (java.util.logging)
Change to 'Long' (java.lang)
Create field 'Log'

Import android.util.Log;

Let’s run it.

Oops, I get this error:

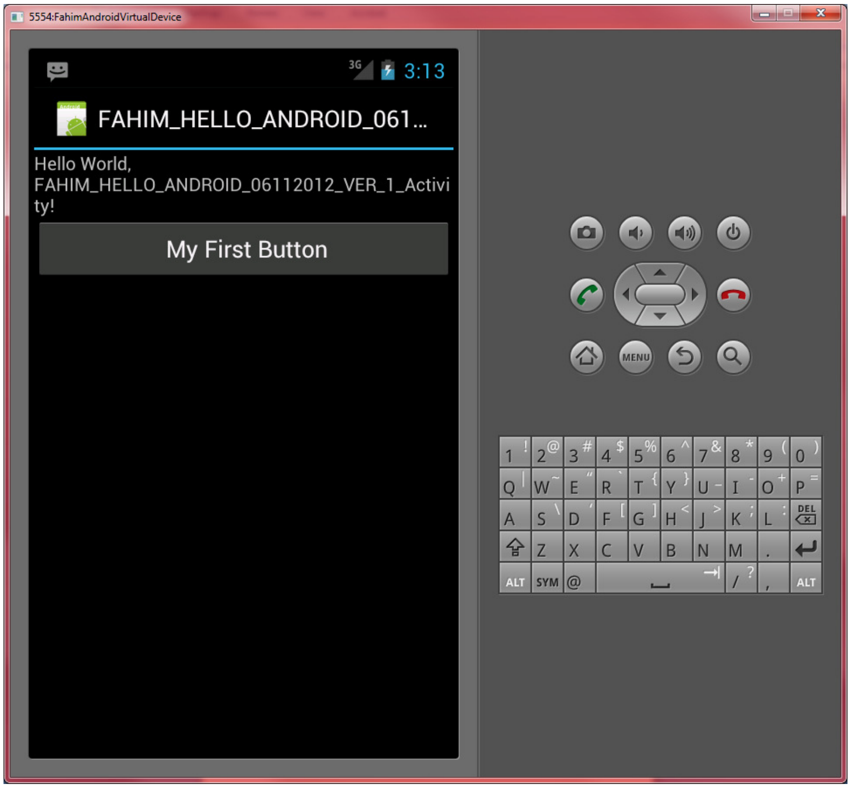


Your project contains error(s), please fix them before running your application.

I forgot to add

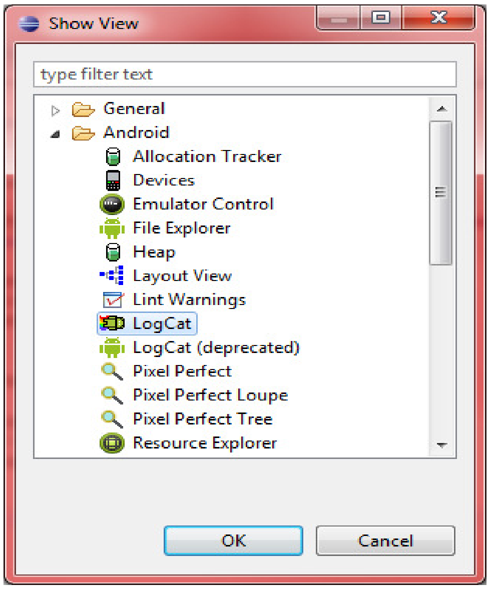
Import android.view.View;

Now Run it.

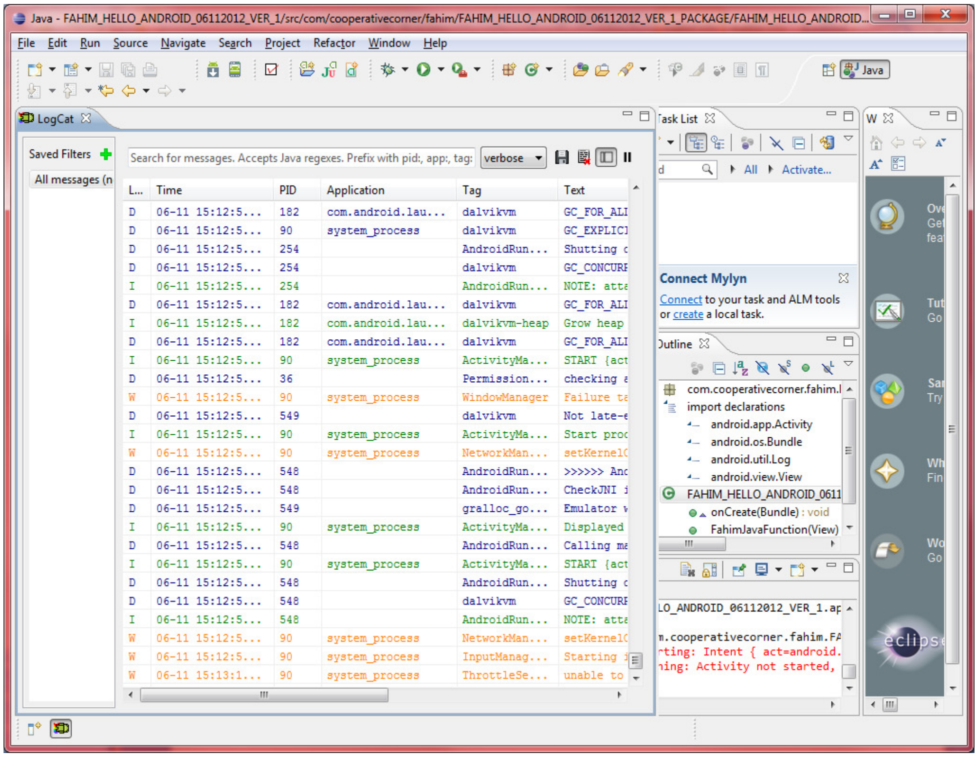


Hello World, FAHIM_HELLO_ANDROID_06112012_VER_1_Activity!

While this is running, lets Open LogCat



This is all, I see as below. So what is in there for me? We will dig more into it, for now, I want to show you that your code will reflect in this debug utility when you click on button.



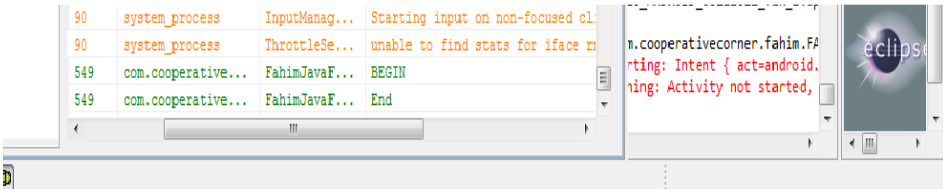
Java - FAHIM_HELLO_ANDROID_06112012_VER_1/src/com/cooperativecorner/fahim/FAHIM_HELLO_ANDROID_06112012_VER_1_PACKAGE/FAHIM_HELLO_ANDROID...
File Edit Run Source Navigate Search Project Refactor Window Help
LogCat
Saved Filters
All messages (n
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag:
verbose
Task List
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
com.cooperativecorner.fahim.l
import declarations
android.app.Activity
android.os.Bundle
android.util.Log
android.view.View
FAHIM_HELLO_ANDROID_0611
onCreate(Bundle) : void
FahimJavaFunction(View)
Java

L...	Time	PID	Application	Tag	Text
D	06-11 15:12:5...	182	com.android.lau...	dalvikvm	GC_FOR_ALI
D	06-11 15:12:5...	90	system_process	dalvikvm	GC_EXPLICI
D	06-11 15:12:5...	254		AndroidRun...	Shutting c
D	06-11 15:12:5...	254		dalvikvm	GC_CONCURR
I	06-11 15:12:5...	254		AndroidRun...	NOTE: atta
D	06-11 15:12:5...	182	com.android.lau...	dalvikvm	GC_FOR_ALI
I	06-11 15:12:5...	182	com.android.lau...	dalvikvm-heap	Grow heap
D	06-11 15:12:5...	182	com.android.lau...	dalvikvm	GC_FOR_ALI
I	06-11 15:12:5...	90	system_process	ActivityMa...	START {act
D	06-11 15:12:5...	36		Permission...	checking a
W	06-11 15:12:5...	90	system_process	WindowManager	Failure ta
D	06-11 15:12:5...	549		dalvikvm	Not late-e
I	06-11 15:12:5...	90	system_process	ActivityMa...	Start proc
W	06-11 15:12:5...	90	system_process	NetworkMan...	setKernelC
D	06-11 15:12:5...	548		AndroidRun...	>>>>>> And
D	06-11 15:12:5...	548		AndroidRun...	CheckJNI i
D	06-11 15:12:5...	549		gralloc_go...	Emulator w
I	06-11 15:12:5...	90	system_process	ActivityMa...	Displayed
D	06-11 15:12:5...	548		AndroidRun...	Calling ma
I	06-11 15:12:5...	90	system_process	ActivityMa...	START {act
D	06-11 15:12:5...	548		AndroidRun...	Shutting c
D	06-11 15:12:5...	548		dalvikvm	GC_CONCURR
I	06-11 15:12:5...	548		AndroidRun...	NOTE: atta
W	06-11 15:12:5...	90	system_process	NetworkMan...	setKernelC
W	06-11 15:12:5...	90	system_process	InputManag...	Starting i
W	06-11 15:13:1...	90	system_process	ThrottleSe...	unable to

```
LO_ANDROID_06112012_VER_1.ap

n.cooperativecorner.fahim.FA
rting: Intent { act=android.
ning: Activity not started,
```

Let’s click the button now.

Do you see what I see?



90	system_process	InputManag...	Starting input on non-focused cl
90	system_process	ThrottleSe...	unable to find stats for iface rm
549	com.cooperative...	FahimJavaF...	BEGIN
549	com.cooperative...	FahimJavaF...	End

```
n.cooperativecorner.fahim.FA
rting: Intent { act=android.
ning: Activity not started,
```

KOOOOL. Our click is responding.

Let’s go and modify the content of Textview during runtime, through Java code.

Add this line of code to Textview node in main.xml file.

```
android:id="+@id/FahimTextView"
```

As soon as you save it, it will magically add the following code to the R.java,

```
public static final class id
{
        public static final int FahimTextView=0x7f050000;
}
```

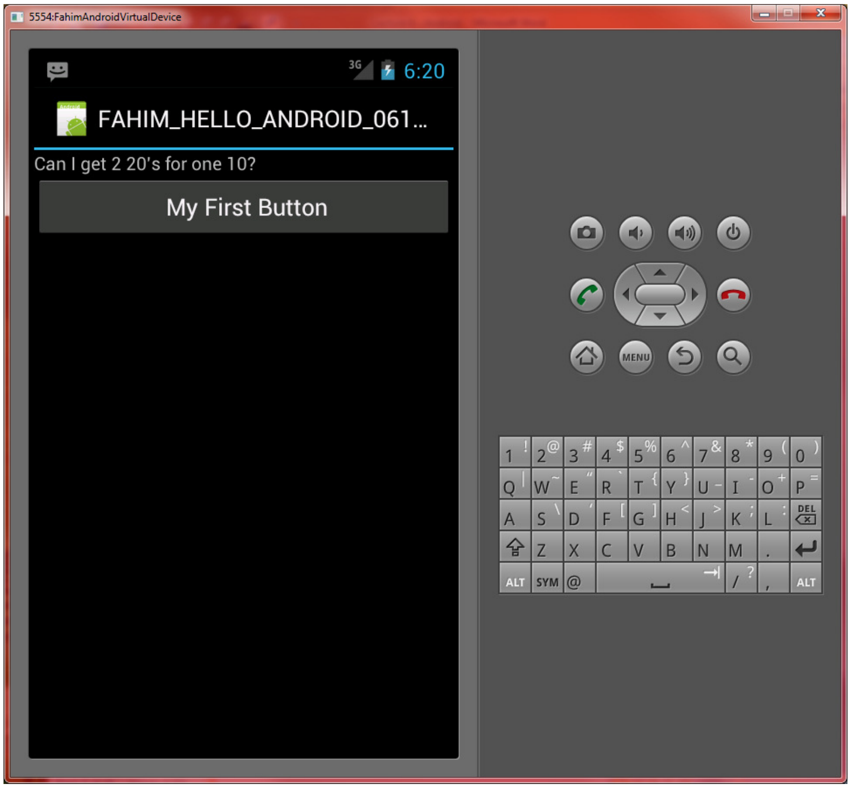
You are getting it. Aren't you?

Let's reference it in Java code. Add the following code in between Log statement in FahimJavaFunction.

```
TextView FahimTextView = (TextView)this.findViewById(R.id.FahimTextView);
                    FahimTextView.setText("Can I get 2 20's for one 10?");
```

Run it.

You should see something like as below screen, once you click on button. If you don't see or it does not work, then it is your problem, not mine. ☺



Can I get 2 20's for one 10?

********************* OUR CODE WORKED*******************

I CAN TELL MY FRIENDS, I AM AN ANDROID APPLICATION DEVELOPER AS I FINALLY WROTE A CODE OF MY OWN.

Let's Now Understand How All Pieces Of Code Really Work

PORT NUMBER:

This is simply TCP/IP port number, very similar to port 80 when your web service runs on it.

If we were to run two separate emulators, like honeycomb tablet in conjunction with Ginger Bread, the port number help identify between these two. Port number also serves as Developer Phone number so text messages, for example, can be sent back and forth. Therefore, two phones can be simulated in Android IDE for app development testing.

What exactly did we do in Java code?

Remember in HTML webpage development, where you use javascript and then call in using OnClick(). Same is true here. What we did was that we created a simple onclick handler in our code.

What code being called when you hit the button with mouse?

Close your eyes and think for a while.

Yes, your guess is right. That is **main.xml**, that is being called at first to draw your User interface.

Check your code out, you would see that you are telling them that call FahimJavaFunction onClick event.

Then you added a line of code into TextView node as follows:

@+id/FahimTextView

What does it tell you, that it could be doing?

Give it a thought; I am sure you can figure it out.

Remember, as soon as you added this line, what happened? When you saved it.

It created a new entry in R.java file as follows:

```
public static final class id {
    public static final int FahimTextView=0x7f050000;
  }
```

Yes, you got that right. It is HexDec address of integer. Now think of HTML code, where if you need to reference one of the objects, you use elementbyId, and HTML will return a handle to the javascript engine. That is exactly we did in java code function using different syntax (who cares if it was different syntax, right?).

In java code, you added

TextView FahimTextView= (TextView)**this**.findViewById(R.id.*FahimTextView*);

We are just returning a Textview from function call here. It is bringing back a pointer in memory to an object. It uses this integer value find a reference by id in this activity (.this),

Also, since view is an abstract type so we had to cast it specifically.

With this done, FahimTextView is a variable of datatype TextView.

FahimTextView.setText("Can I get 2 20's for one 10?");

This line of action looks similar to Visual Studio label control right?

But note, it behaves very different though. So don't take it exactly like label control.

Even you have seen some similarities between Android-Java and HTML-javascript, but here is a difference or one of the differences that when we specify javascript function in html, we specify open and close parenthesis, but in android world, **we don't.**

9. STUDY RESOURCES

During lecture, I will give you name of some good books for you out there. I will also discuss my own book that I am in process of composing at the present time. My book as a text book will be available by **spring 2013**.

Here are some worth resources for you to go through:

1. http://www.netmite.com/android/mydroid/development/pdk/docs/index.html
2. https://developers.google.com/android/
3. http://source.android.com/compatibility/index.html
4. http://www.helloandroid.com/tutorials
5. http://www.vogella.com/articles/Android/article.html
6. http://www.youtube.com (of course).