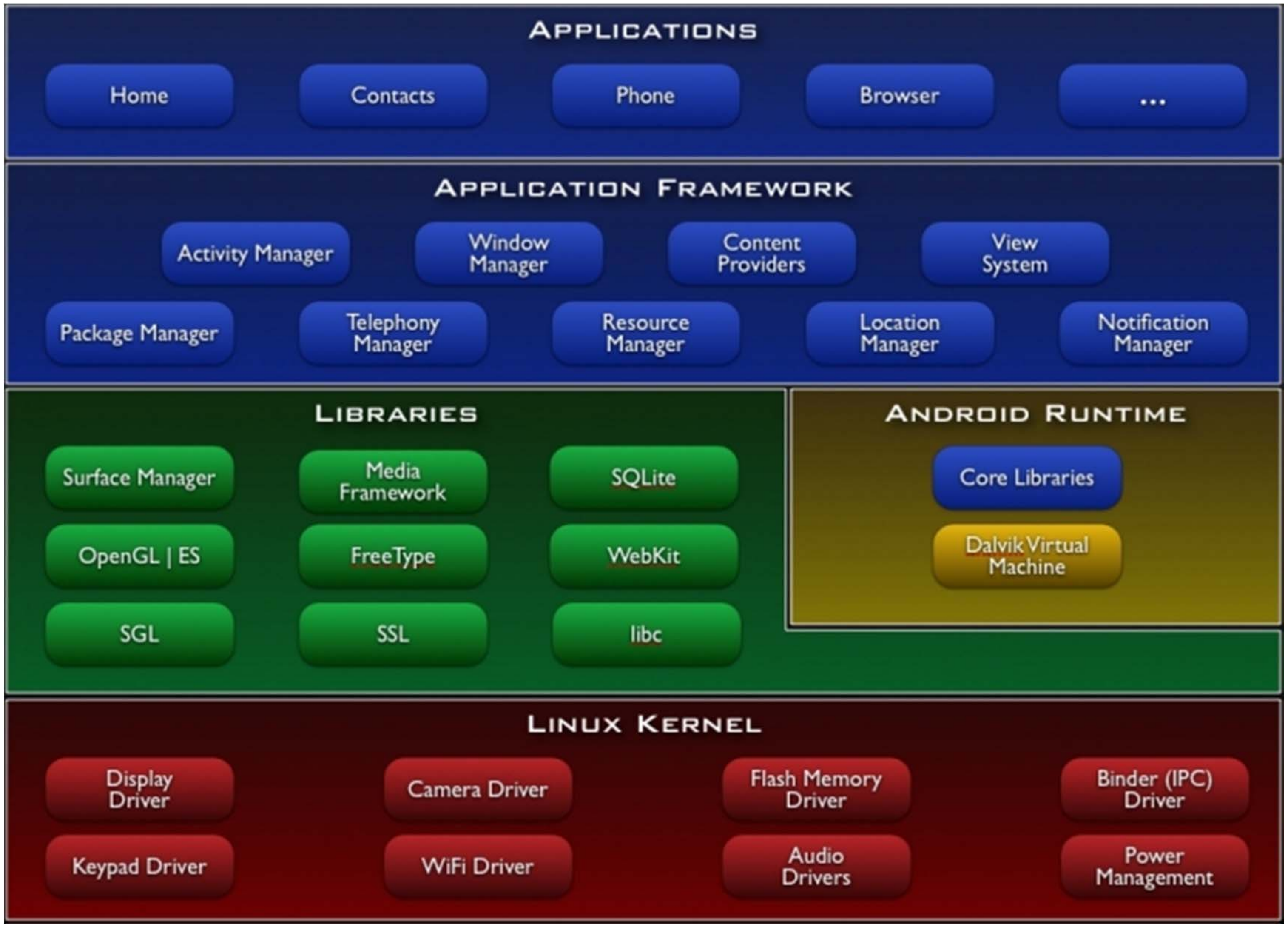
Android Programming
Lecture 1

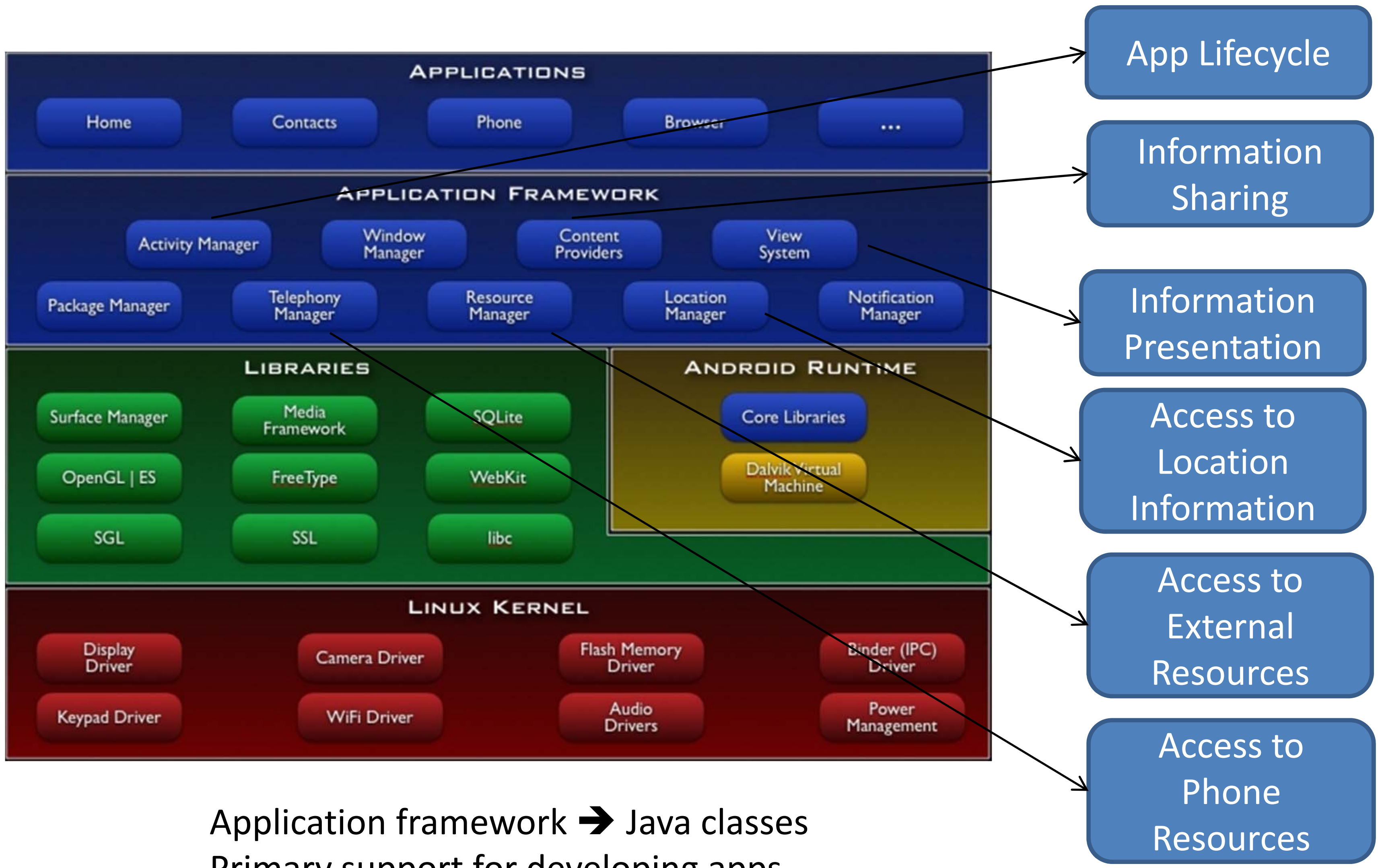
9/2/2011

Android Software Architecture



http://developer.android.com/guide/basics/what-is-android.html

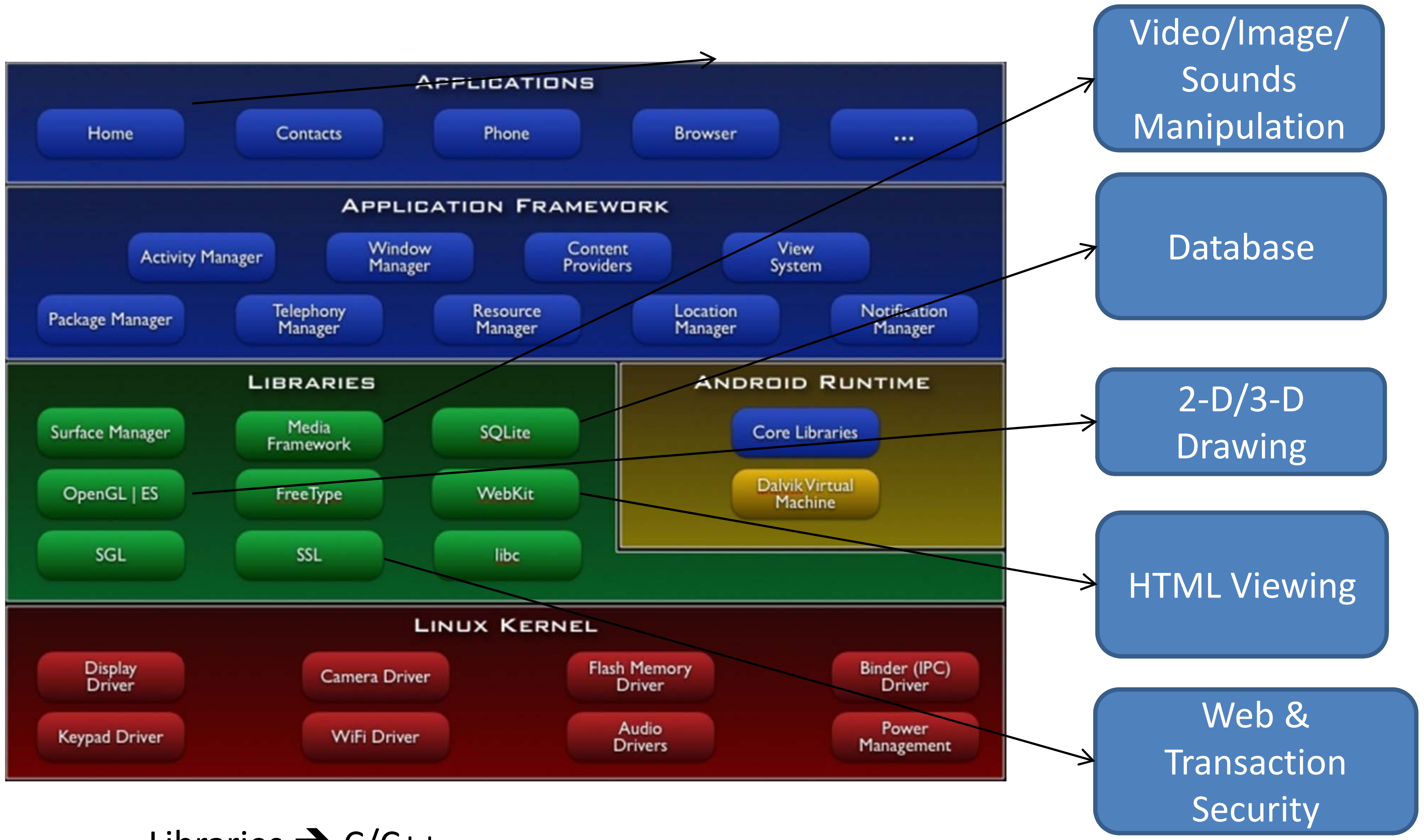
Android Software Architecture



Application framework ➔ Java classes

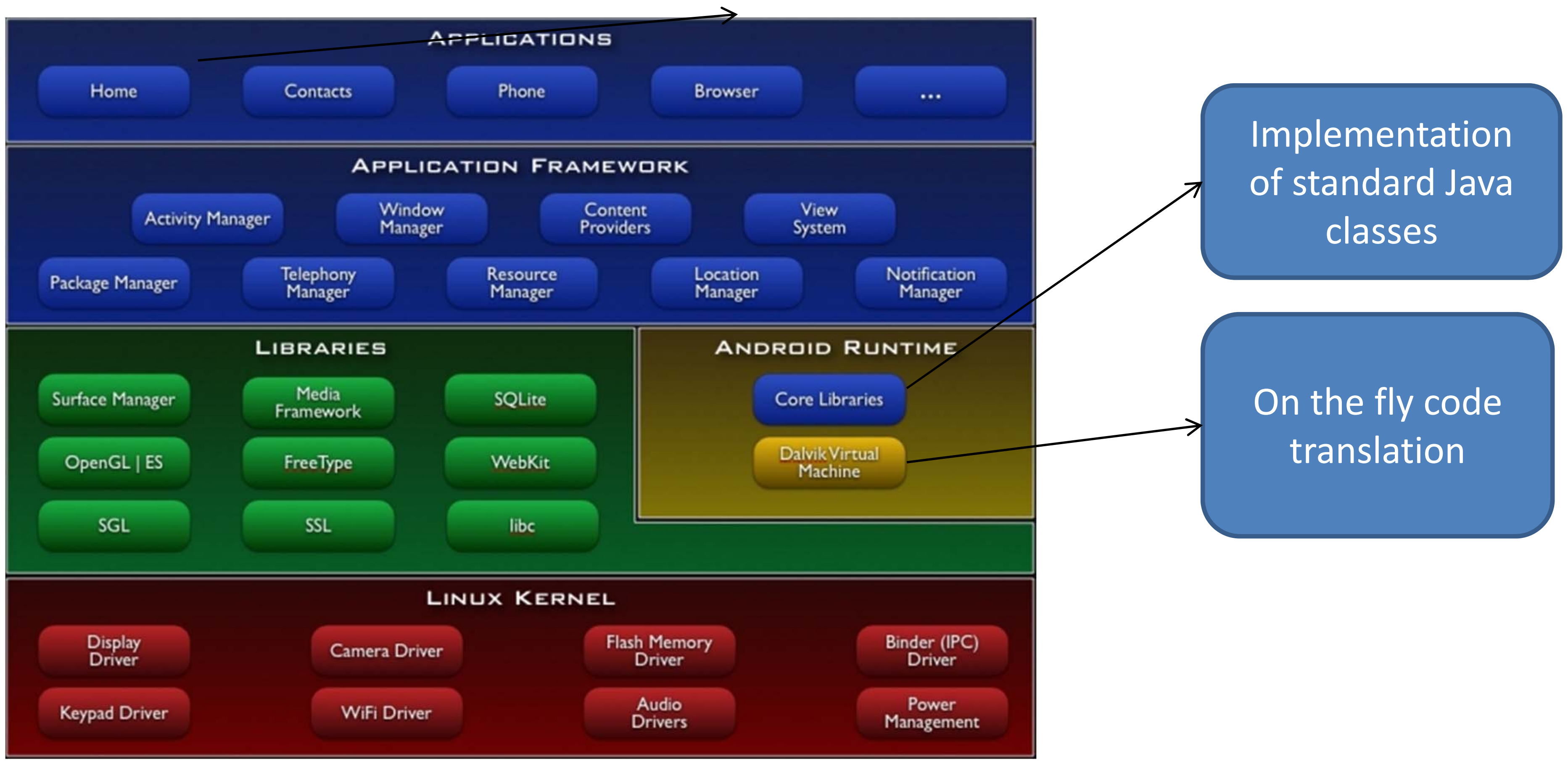
Primary support for developing apps

Android Software Architecture



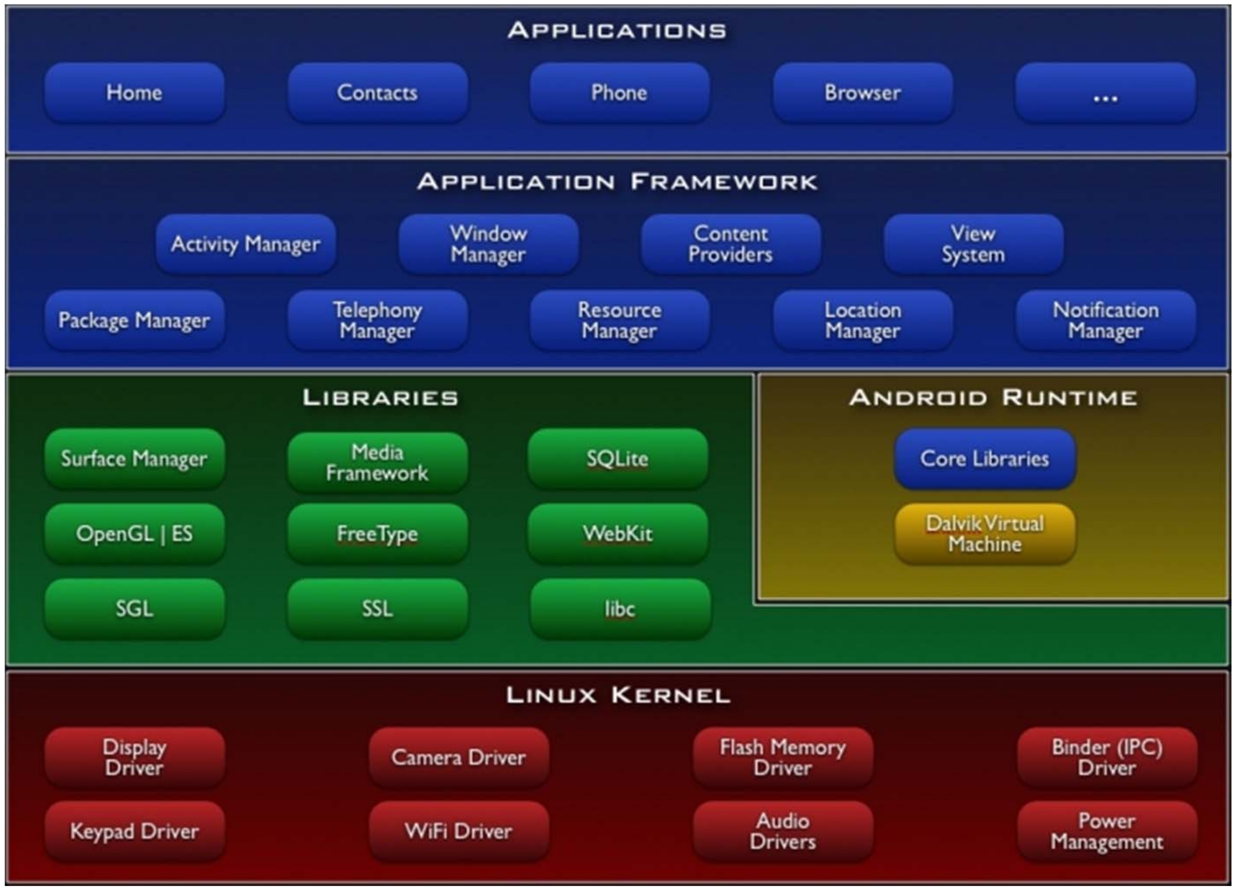
Libraries ➔ C/C++
Implemented functionality supporting application framework

Android Software Architecture



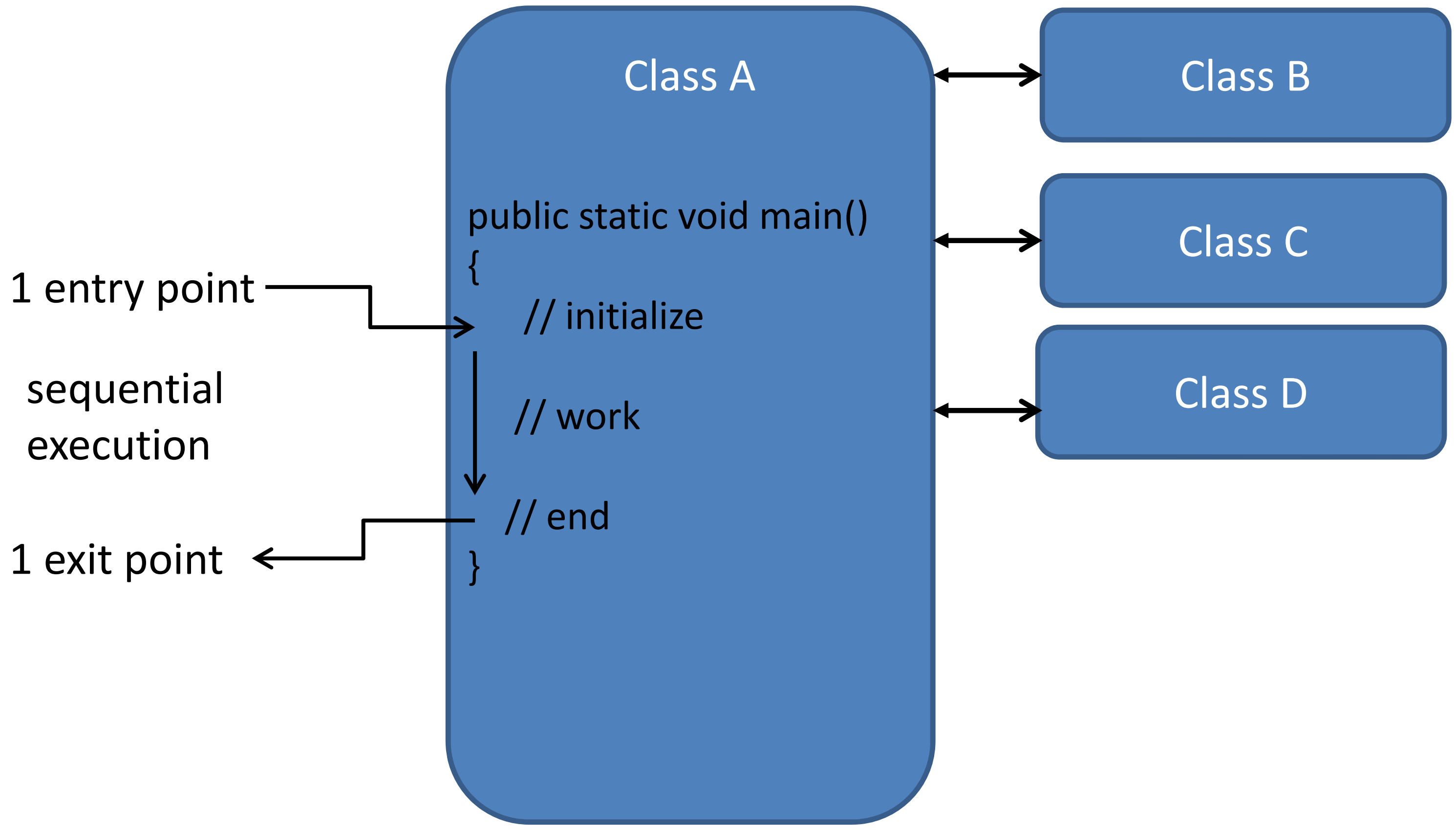
Android Runtime ➔ C/C++
Support program execution at runtime

Android Software Architecture

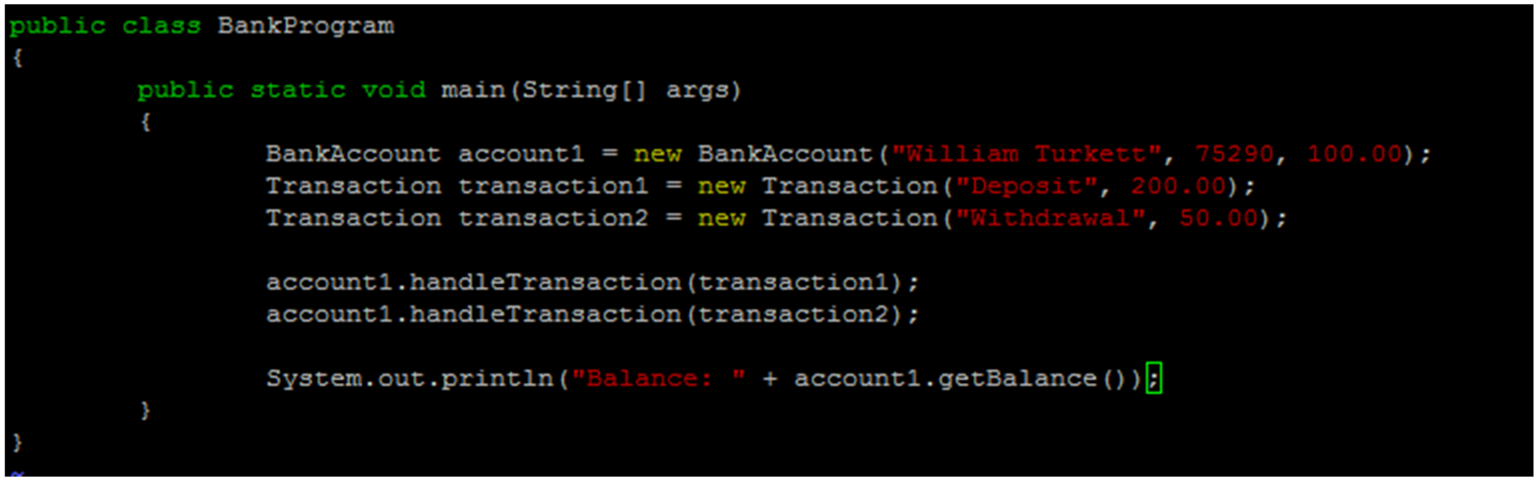


Linux Kernel ➔ C/C++
Interface with Hardware
OS Services – memory management, scheduling, …

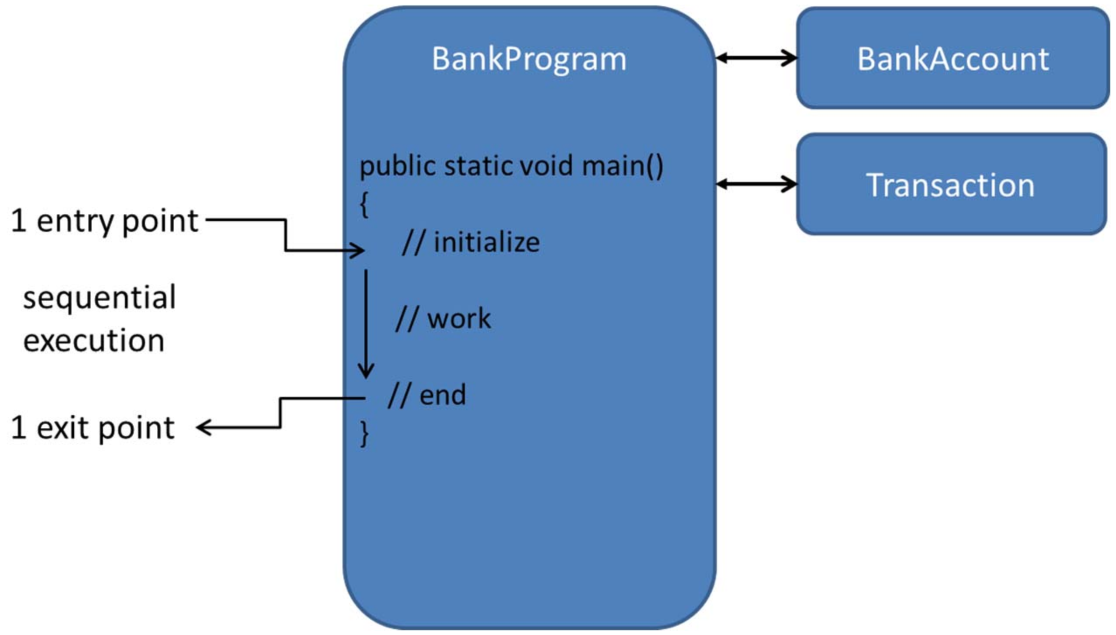
Traditional Java Application



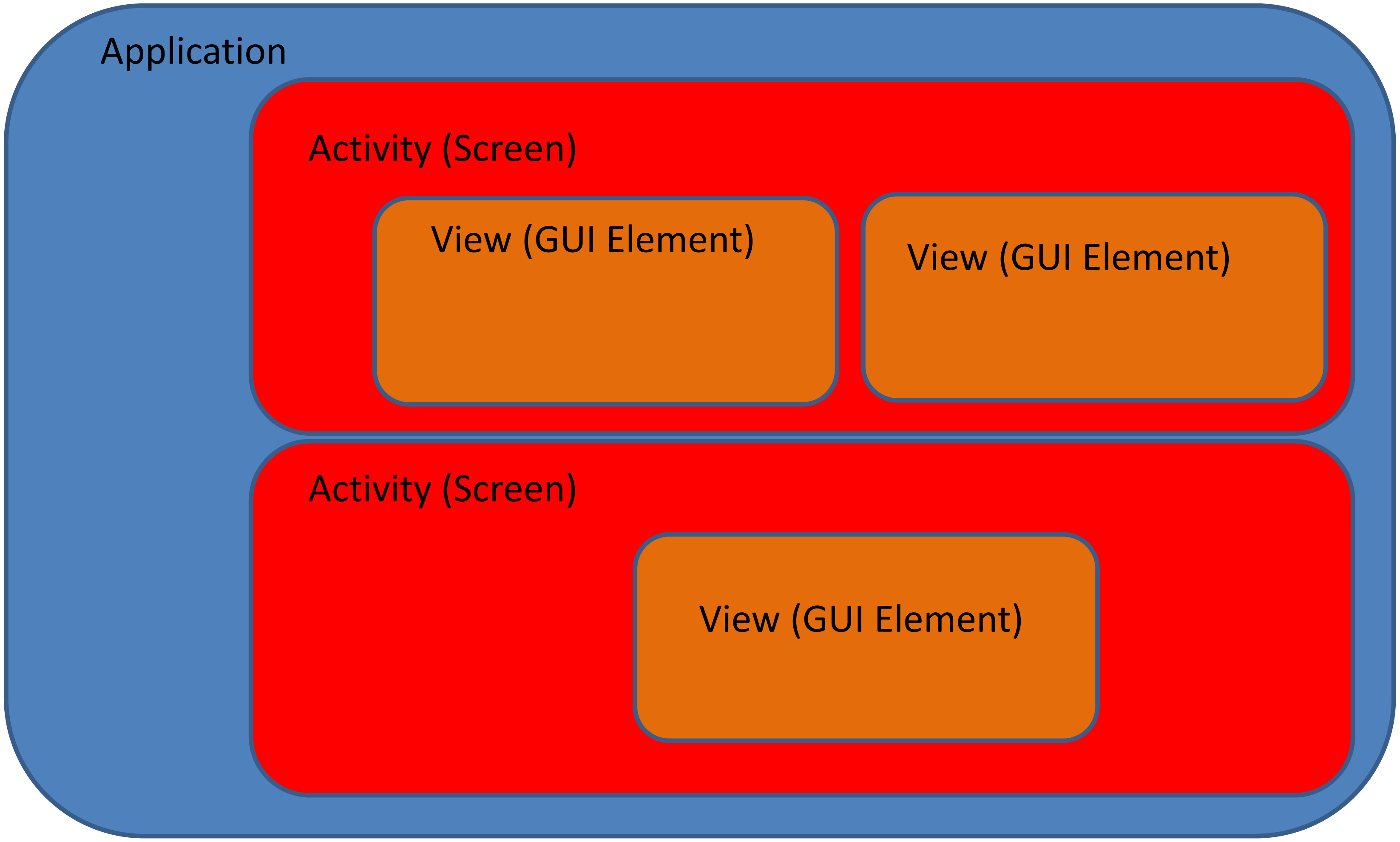
Traditional Java Application



```
public class BankProgram
{
        public static void main(String[] args)
        {
                BankAccount account1 = new BankAccount("William Turkett", 75290, 100.00);
                Transaction transaction1 = new Transaction("Deposit", 200.00);
                Transaction transaction2 = new Transaction("Withdrawal", 50.00);

                account1.handleTransaction(transaction1);
                account1.handleTransaction(transaction2);

                System.out.println("Balance: " + account1.getBalance());
        }
}
```

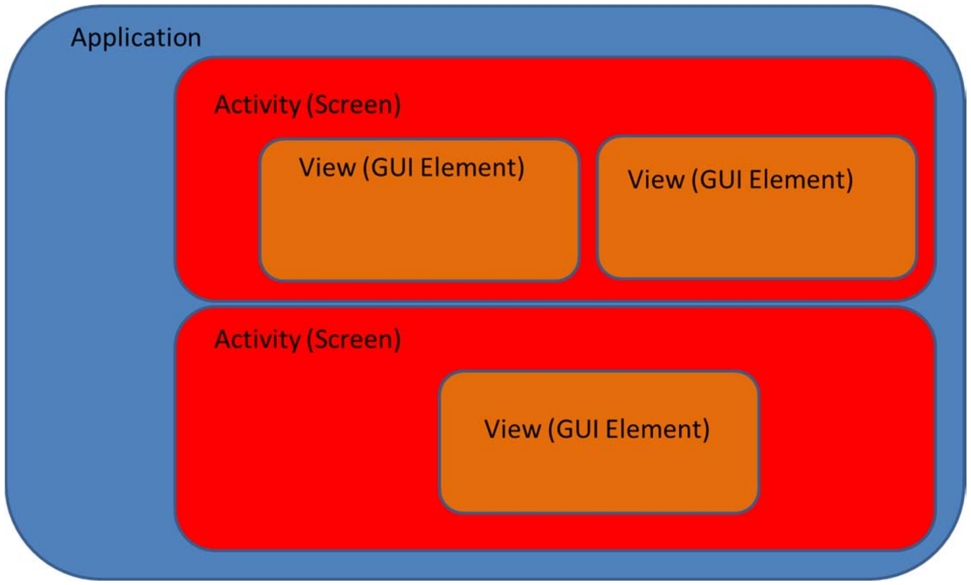


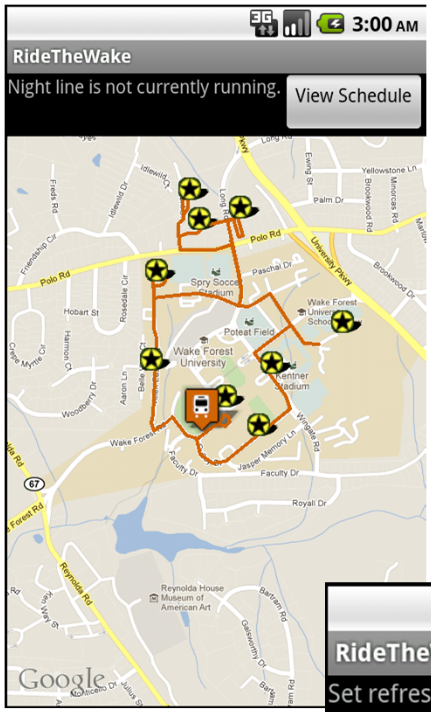
Simple Android Application: Design



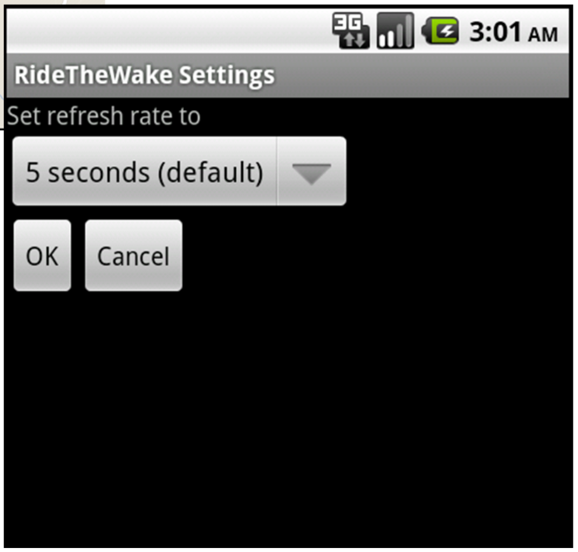
An application with two screens. Each screen is composed of 1 or more GUI elements.

Simple Android Application: Design and Example





Night line is not currently running.



Set refresh rate to

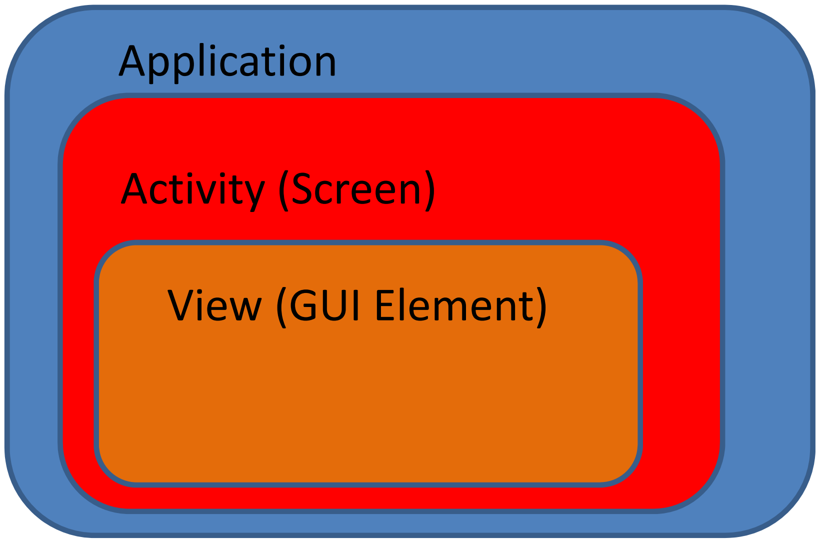
5 seconds (default)

An application with two screens.

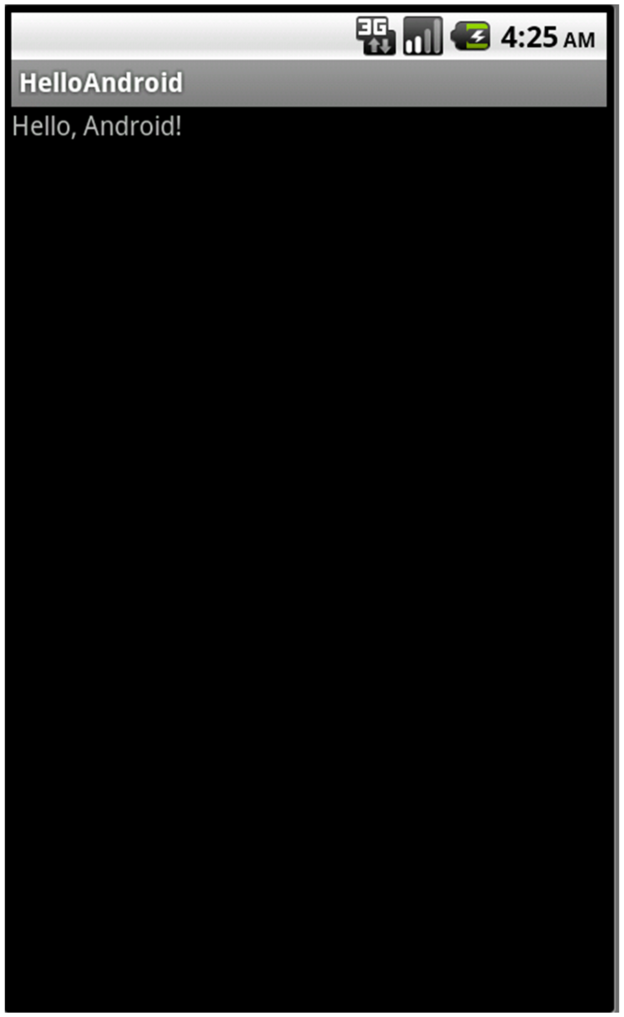
- Main map screen
- Settings

Each screen is composed of several GUI elements.

Simple Android Application: Our Initial Goal



Our goal: An Application with one Activity with one View (a big textbox)



Hello, Android!

Application Priority

- The Android OS allows multiple applications to be executing at a time
- To service an application, the OS can reclaim resources by killing another application.
- Lowest priority = most likely to be killed
- Within a level ➔ least recently seen = first to be killed

<u>Priority Hierarchy</u>

Active applications

Visible applications

Service applications

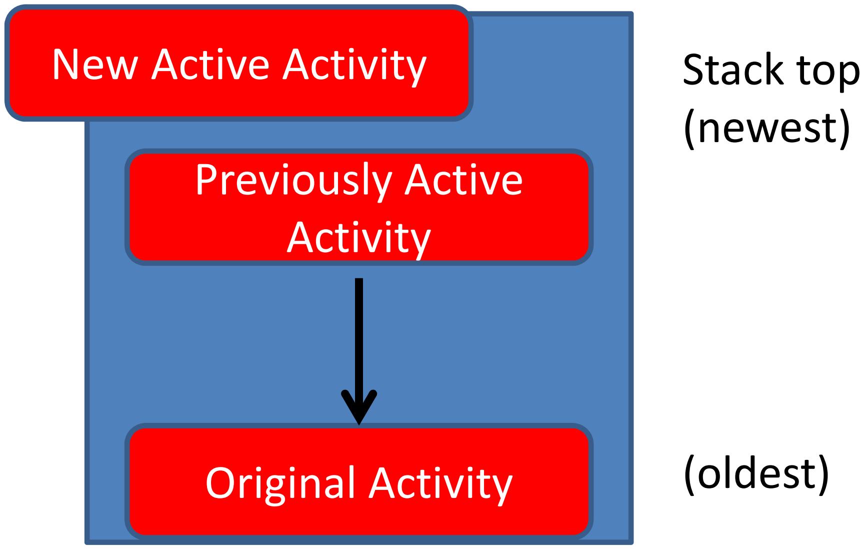
Background applications

“Empty” (completed) applications

Activity States

- Within an application, there can be multiple Activities (screens)

- Activities are maintained in a stack



Activity States

Active

Foreground – receiving input

Paused

Visible but obscured

Stopped

No longer visible, still in memory

Inactive

Not visible, not in memory (terminated)

**Activities can be killed, just like applications, based on priority; states above are in priority order*

Applications and Activities

Applications and Activities are just Java classes

android.app.Application

android.app.Activity

We will interact with them using *member functions*

Initially, we will use a default *Application* setup and focus on building *Activities.*

When an *Application* is selected on the device, it will trigger a first *Activity* to be instantiated.

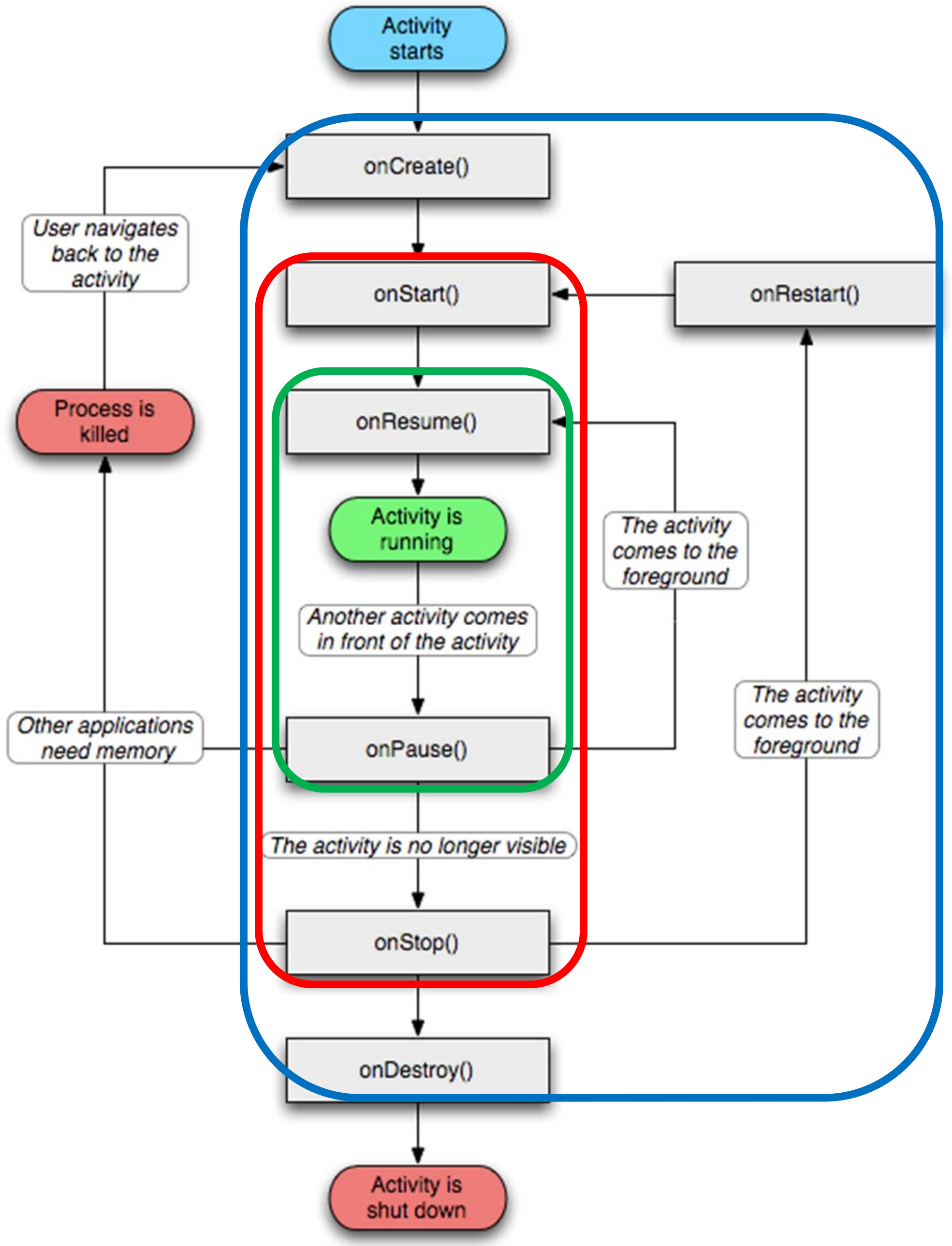
Android App Lifecycle

As an Activity moves through its possible different states, functions are automatically called on the Activity, triggering different parts of our code.

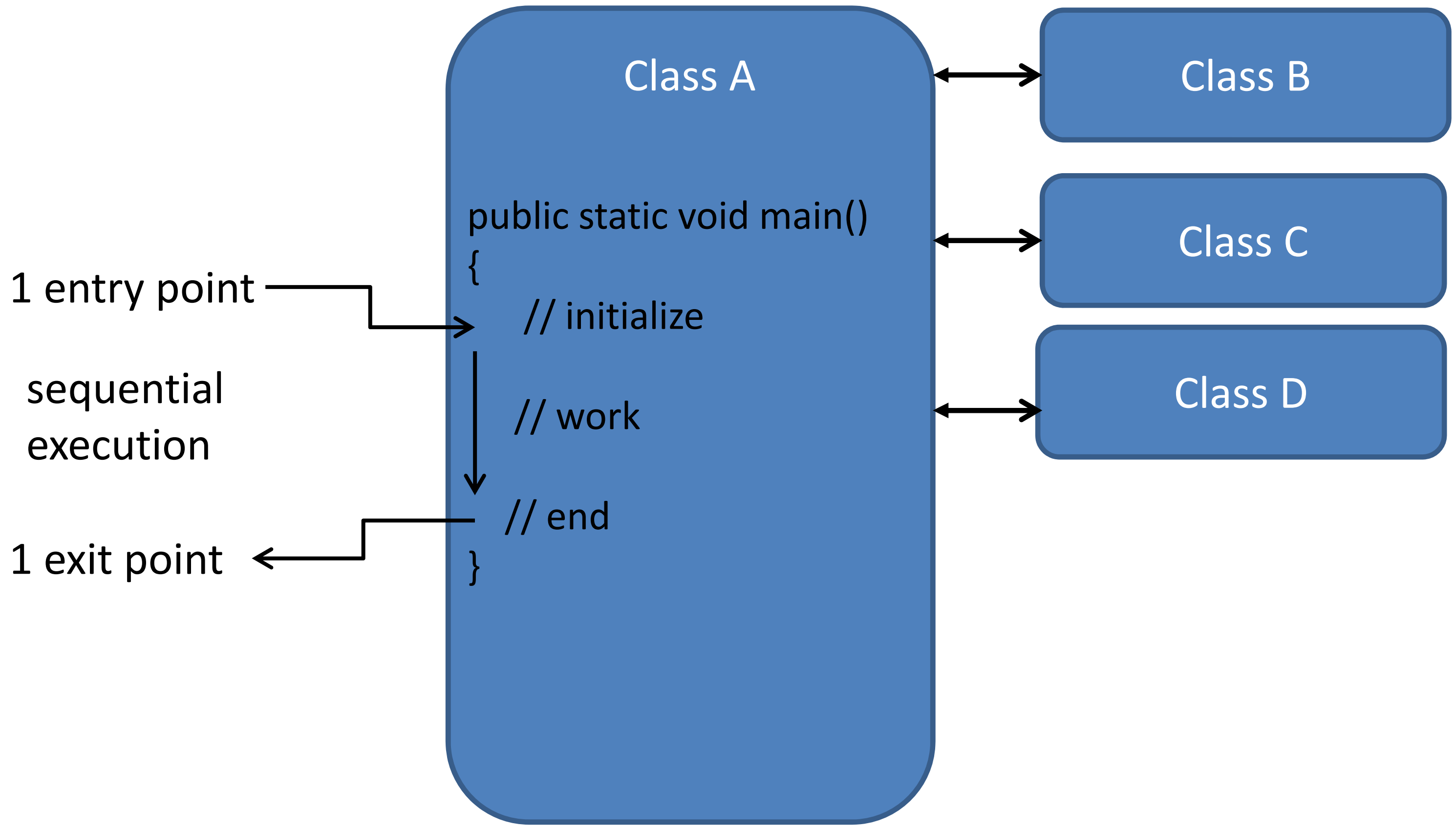
[Remember this happens for every screen!]

The first such function is called *onCreate* and is triggered when an Activity is first requested.

Active (state: active)
Visible (state: active or paused)
Alive (active, paused, stopped)

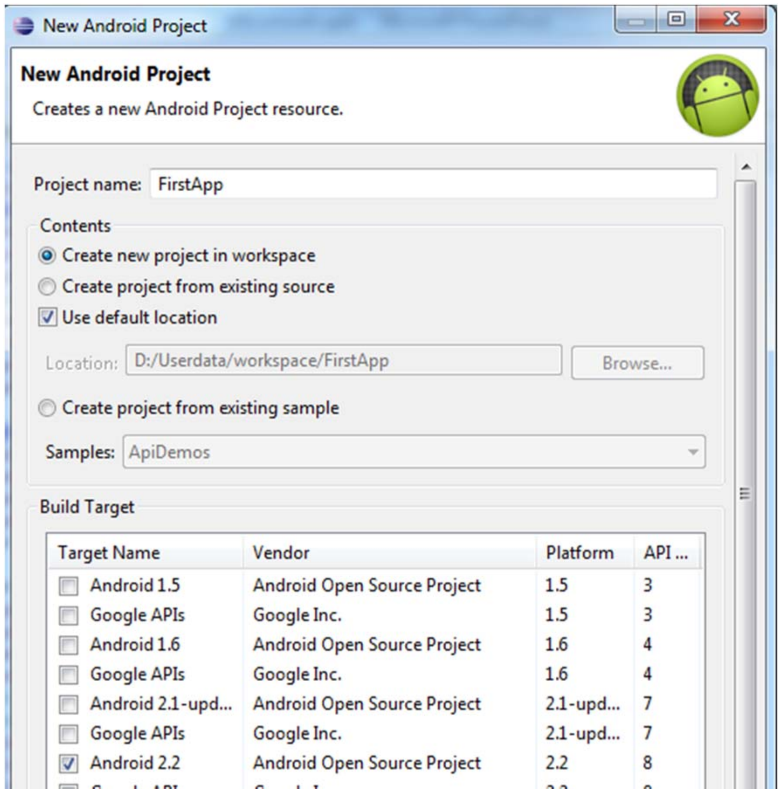

Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
The activity comes to the foreground
Other applications need memory
onPause()
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Traditional Java Application



Creating a first app

1. Create a new Android project (a collection of source code and resources for the app) from the Eclipse file menu
2. Choose a project name (can be anything)
3. Application specifics:
   1. Target platform
   2. Application name
   3. Package name
   4. Initial activity to launch
   5. Absolute minimum platform
4. Finish



New Android Project

New Android Project

Creates a new Android Project resource.

Project name: FirstApp

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

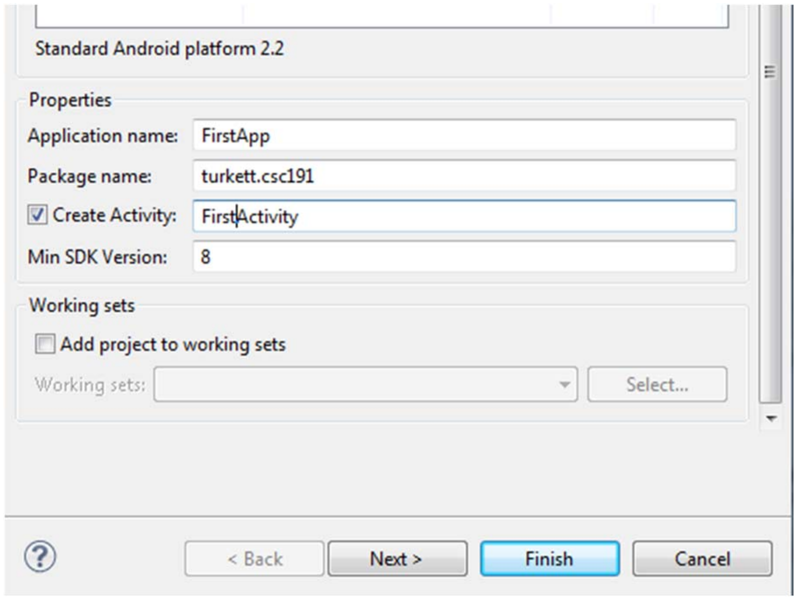
Location: D:/Userdata/workspace/FirstApp — Browse...

○ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API ...
☐ Android 1.5	Android Open Source Project	1.5	3
☐ Google APIs	Google Inc.	1.5	3
☐ Android 1.6	Android Open Source Project	1.6	4
☐ Google APIs	Google Inc.	1.6	4
☐ Android 2.1-upd...	Android Open Source Project	2.1-upd...	7
☐ Google APIs	Google Inc.	2.1-upd...	7
☑ Android 2.2	Android Open Source Project	2.2	8



Standard Android platform 2.2

Properties

Application name: FirstApp

Package name: turkett.csc191

☑ Create Activity: FirstActivity

Min SDK Version: 8

Working sets

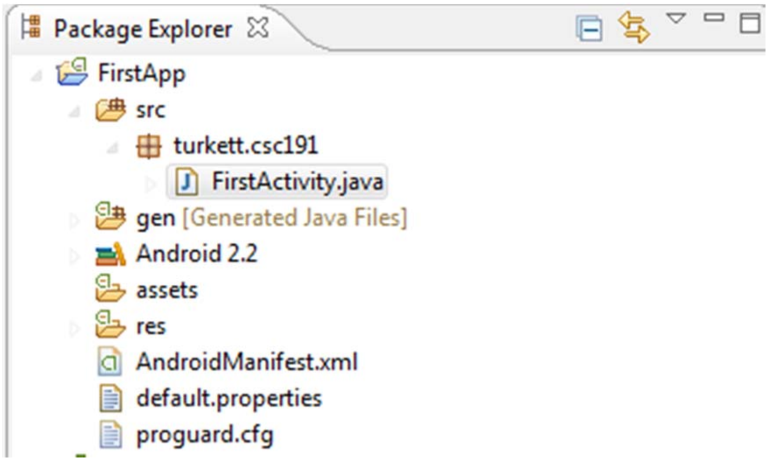
☐ Add project to working sets

Working sets: — Select...

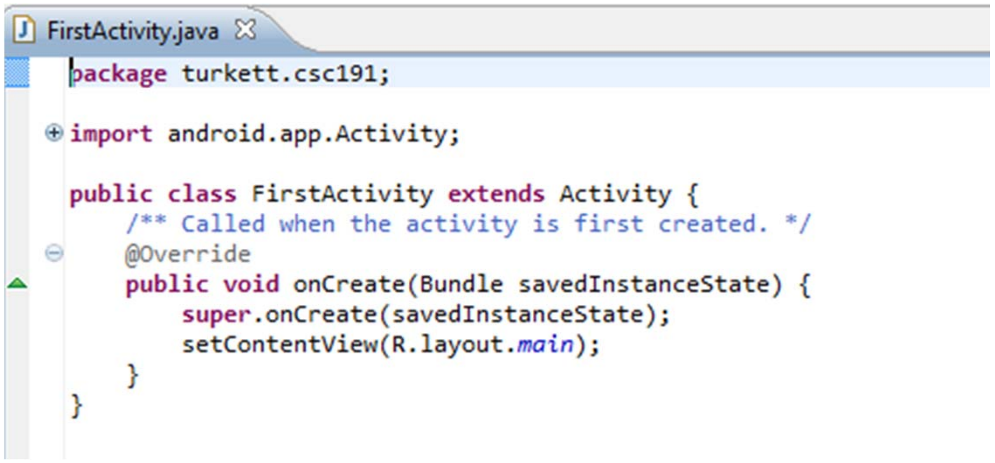
< Back — Next > — Finish — Cancel

Creating a first app

1. Expand the project, src folder, and your chosen package



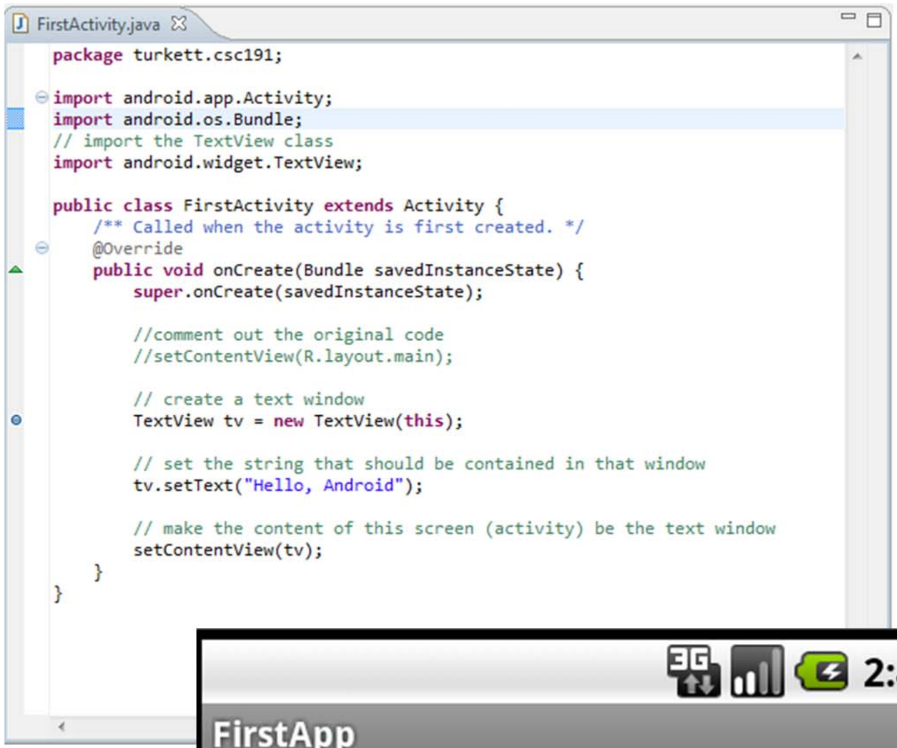
2. Choosing your Activity file will reveal a default implementation of the *onCreate* function
   1. Calls the onCreate of the Activity parent class
   2. Sets the content of this screen to be an XML specified layout *(we'll come back to this)*



```
package turkett.csc191;

import android.app.Activity;

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

Creating a first app

3. Replace pre-generated code with your own TextView code

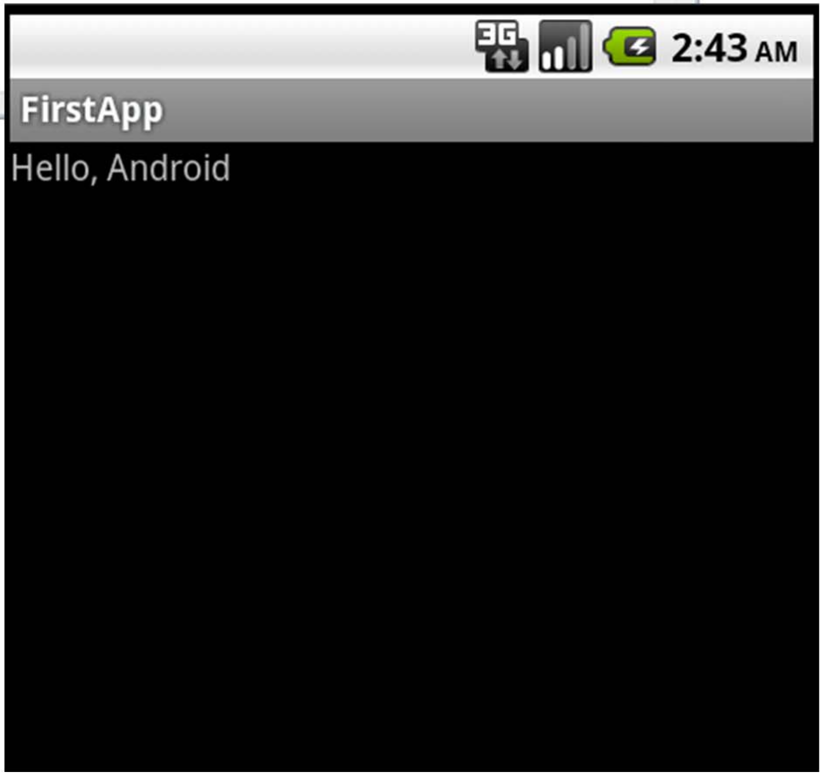


FirstActivity.java

```
package turkett.csc191;

import android.app.Activity;
import android.os.Bundle;
// import the TextView class
import android.widget.TextView;

public class FirstActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //comment out the original code
        //setContentView(R.layout.main);

        // create a text window
        TextView tv = new TextView(this);

        // set the string that should be contained in that window
        tv.setText("Hello, Android");

        // make the content of this screen (activity) be the text window
        setContentView(tv);
    }
}
```

4. Run the app from Eclipse

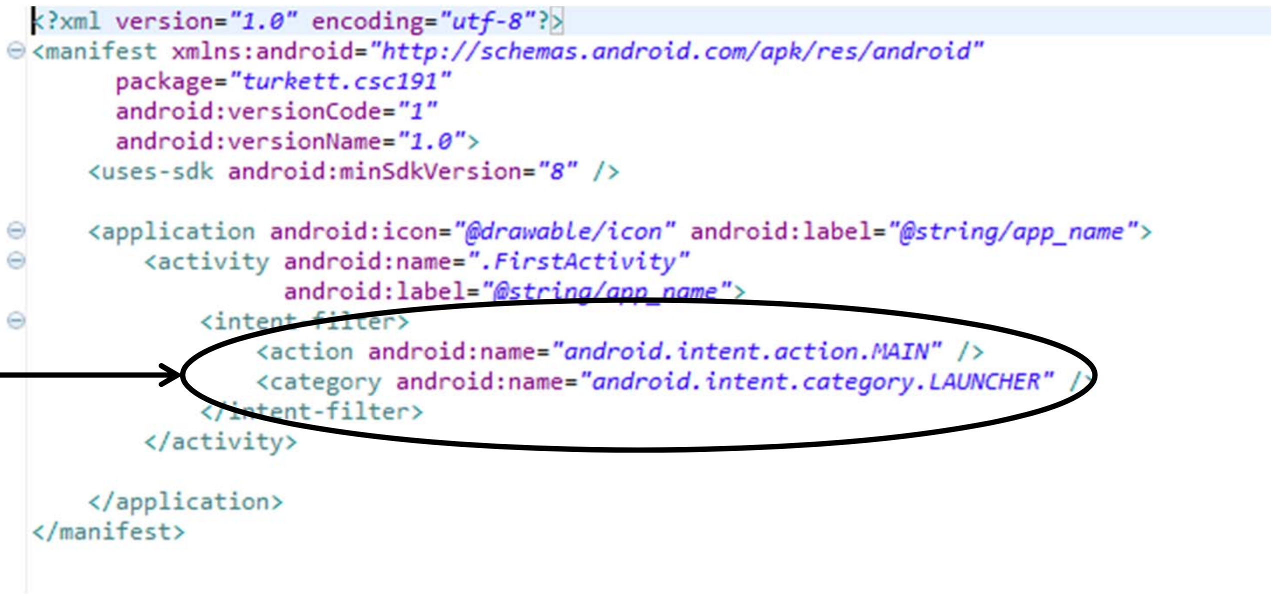
5. Emulator should start, and open your app



```
2:43 AM
FirstApp
Hello, Android
```

Applications and Activities

- How does the Application know the initial Activity to call?
  - Stored in application manifest: AndroidManifest.xml
    - Managed by Eclipse for us

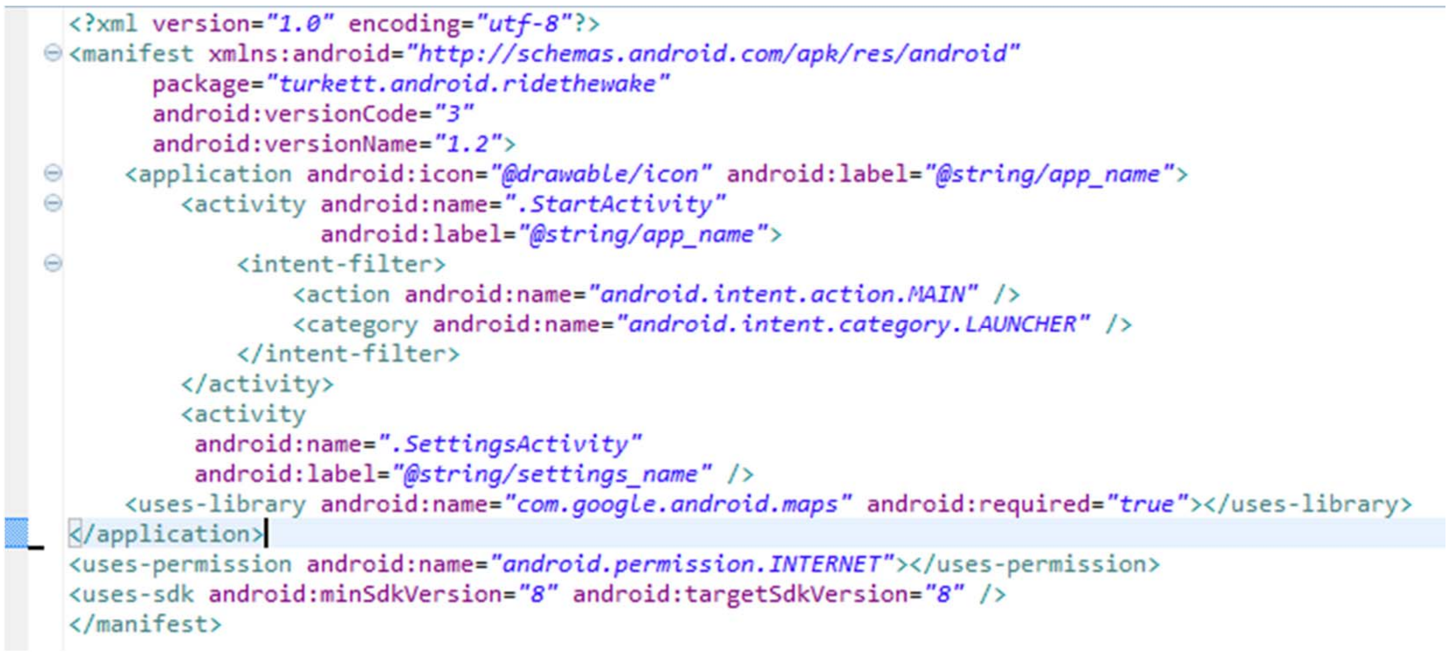


```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="turkett.csc191"
      android:versionCode="1"
      android:versionName="1.0">
    <uses-sdk android:minSdkVersion="8" />

    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".FirstActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
</manifest>
```

Indication that the activity is the first target

Applications and Activities

- A manifest for an Application with two Activity components



```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="turkett.android.ridethewake"
      android:versionCode="3"
      android:versionName="1.2">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".StartActivity"
                  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=".SettingsActivity"
         android:label="@string/settings_name" />
    <uses-library android:name="com.google.android.maps" android:required="true"></uses-library>
</application>
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="8" />
</manifest>
```

Important Java Concepts

- Packages:
  - packages of classes = directories of files
    - Importing in Java
    - Your own
- Inheriting from Activity/super
- Becoming familiar with the Android API
  - http://developer.android.com/reference/packages.html
  - http://developer.android.com/reference/classes.html