

Welcome to Android

# Chapter 2 – Getting Started

Android SDK contains:

- API Libraries
- Developer Tools
- Documentation
- Sample Code

Best development environment is Eclipse with the Android Developer Tool (ADT) plugin which integrates developer tools

# Android Portability

Android applications run within the Dalvik virtual machine

Development Platforms:

Windows (XP, Windows, 7)

Linux

Mac OS 10.4.8 or later (Intel chips only)

# HelloWorld Android Activity

File->New->(Android Project or Other->Android Project)

**New Android Project**

Creates a new Android Project resource.

Project name: HelloWorld

**Contents**

- Create new project in workspace
- Create project from existing source
- Use default location

Location: C:/Documents and Settings/ryandj/workspace/HelloWorld

Create project from existing sample

Samples: ApiDemos

**Build Target**

| Target Name                                     | Vendor                      | Platform | API... |
|---|-----------------------------|----------|--------|
| <input checked="" type="checkbox"/> Android 1.1 | Android Open Source Project | 1.1      | 2      |
| <input type="checkbox"/> Android 1.5            | Android Open Source Project | 1.5      | 3      |
| <input type="checkbox"/> Android 1.6            | Android Open Source Project | 1.6      | 4      |
| <input type="checkbox"/> Android 2.0            | Android Open Source Project | 2.0      | 5      |
| <input type="checkbox"/> Android 2.0.1          | Android Open Source Project | 2.0.1    | 6      |
| <input type="checkbox"/> Android 2.1            | Android Open Source Project | 2.1      | 7      |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 1.5      | 3      |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 1.6      | 4      |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 2.0      | 5      |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 2.0.1    | 6      |
| <input type="checkbox"/> Google APIs            | Google Inc.                 | 2.1      | 7      |

Standard Android platform 1.1

**Properties**

Application name: HelloWorld

Package name: edu.pacificu.cs>HelloWorld

Create Activity: HelloWorld

Min SDK Version: 2

# Skip New Android Test Project

**New Android Project**

**New Android Test Project**  
Creates a new Android Test Project resource.

**Create a Test Project:**

Test Project Name:

**Content**

Use default location

Location:

**Test Target**

Test Target Package:

**Build Target**

| Target Name                            | Vendor                      | Platform | API... |
|--|-----------------------------|----------|--------|
| <input type="checkbox"/> Android 1.1   | Android Open Source Project | 1.1      | 2      |
| <input type="checkbox"/> Android 1.5   | Android Open Source Project | 1.5      | 3      |
| <input type="checkbox"/> Android 1.6   | Android Open Source Project | 1.6      | 4      |
| <input type="checkbox"/> Android 2.0   | Android Open Source Project | 2.0      | 5      |
| <input type="checkbox"/> Android 2.0.1 | Android Open Source Project | 2.0.1    | 6      |
| <input type="checkbox"/> Android 2.1   | Android Open Source Project | 2.1      | 7      |
| <input type="checkbox"/> Google APIs   | Google Inc.                 | 1.5      | 3      |
| <input type="checkbox"/> Google APIs   | Google Inc.                 | 1.6      | 4      |
| <input type="checkbox"/> Google APIs   | Google Inc.                 | 2.0      | 5      |
| <input type="checkbox"/> Google APIs   | Google Inc.                 | 2.0.1    | 6      |
| <input type="checkbox"/> Google APIs   | Google Inc.                 | 2.1      | 7      |

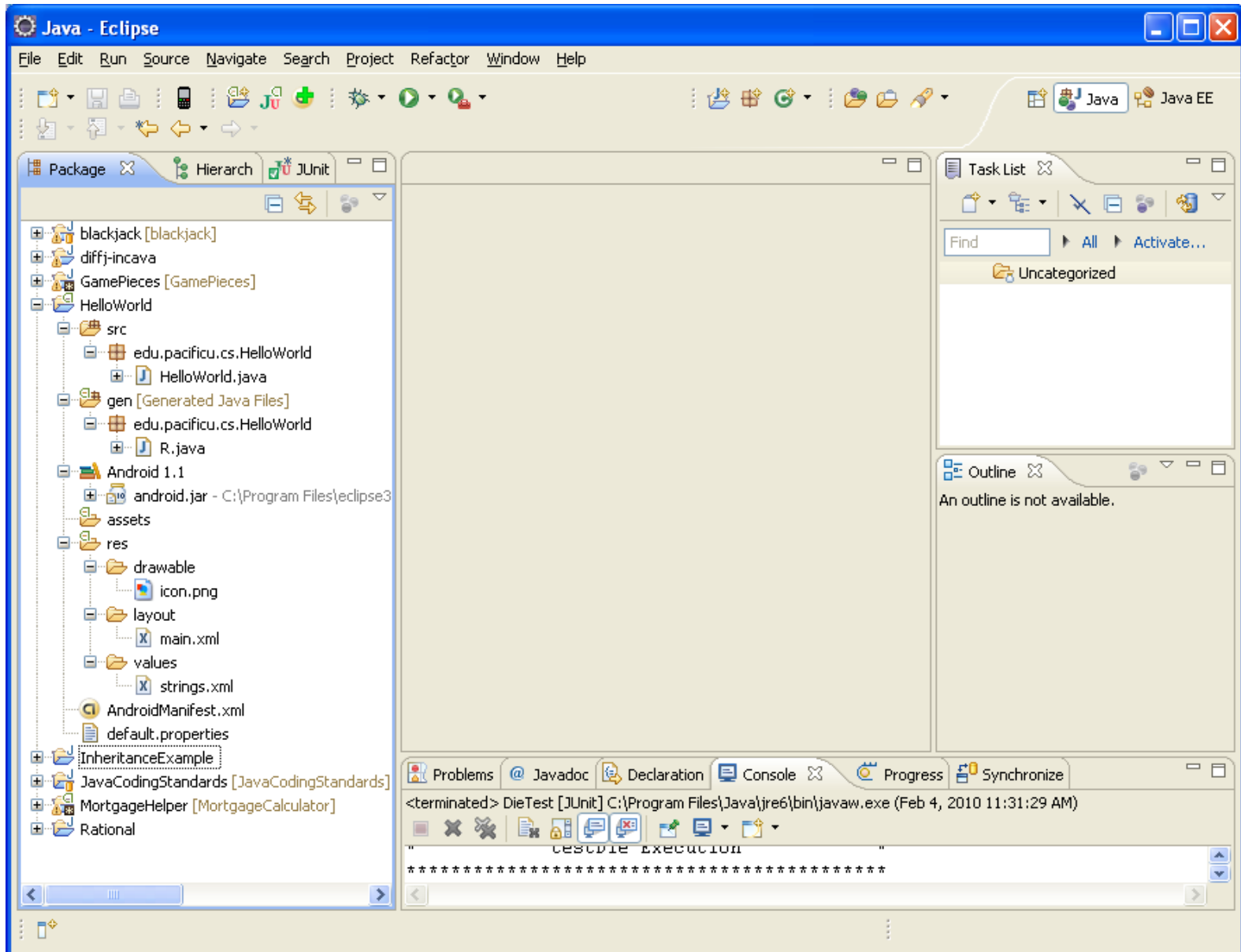
**Properties**

Application name:

Package name:

Min SDK Version:

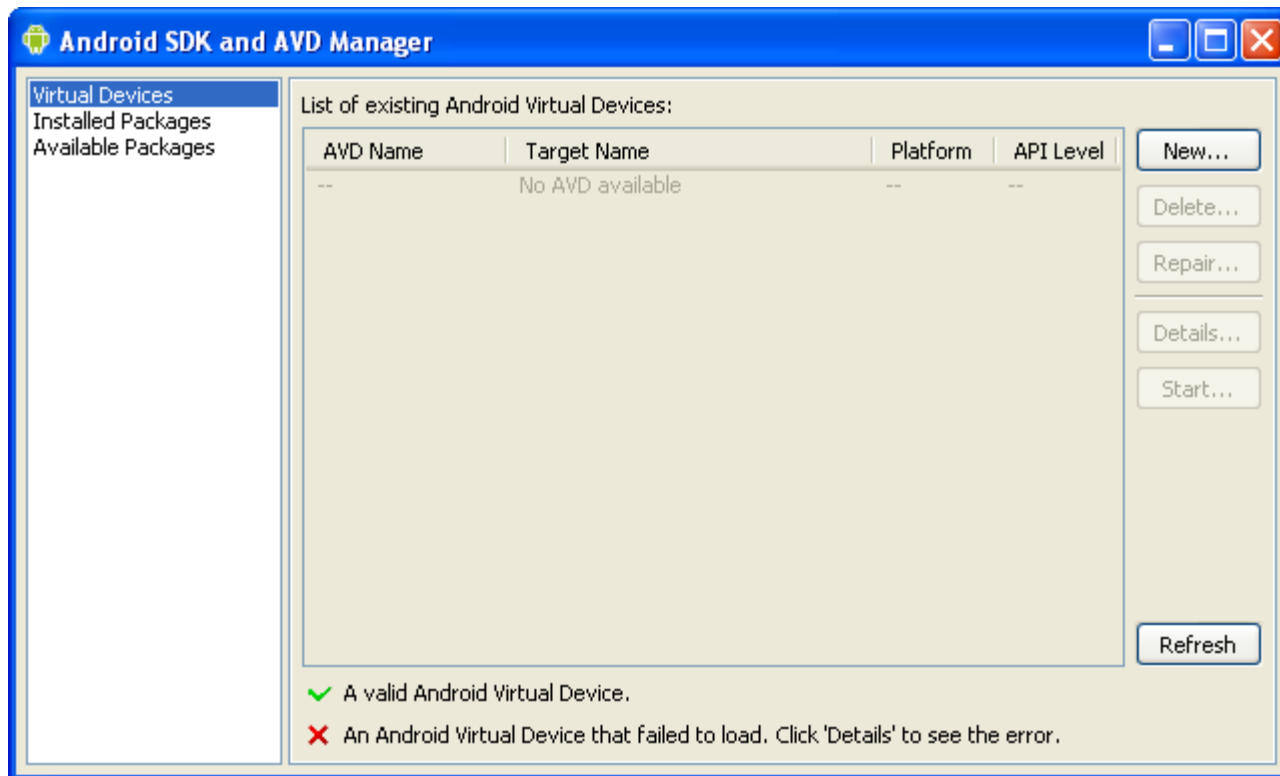
# HelloWorld Android Project



# Running Android Project

Before you can run an Android application, you need to create an Android Virtual Device (AVD)

1. Window->Android SDK and AVD Manager
2. Create a virtual device called **AVD1.1** using the 1.1 Platform
3. Use an SD Card of 32MB for later



# Creating Virtual Device

**Create new AVD**

Name:

Target:

SD Card:

Size:  MIB

File:  Browse...

Skin:

Built-in:

Resolution:

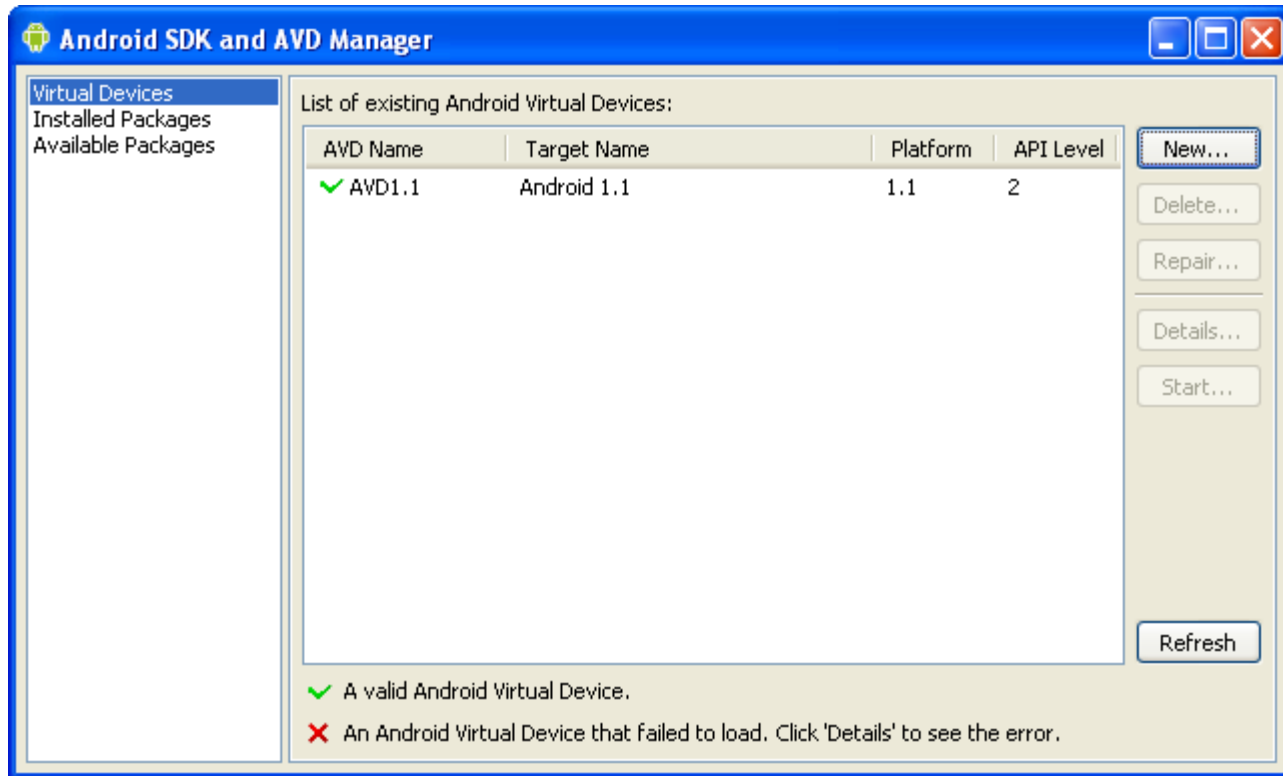
Hardware:

| Property | value |
|----------|-------|
|          |       |
|          |       |
|          |       |
|          |       |
|          |       |
|          |       |
|          |       |
|          |       |

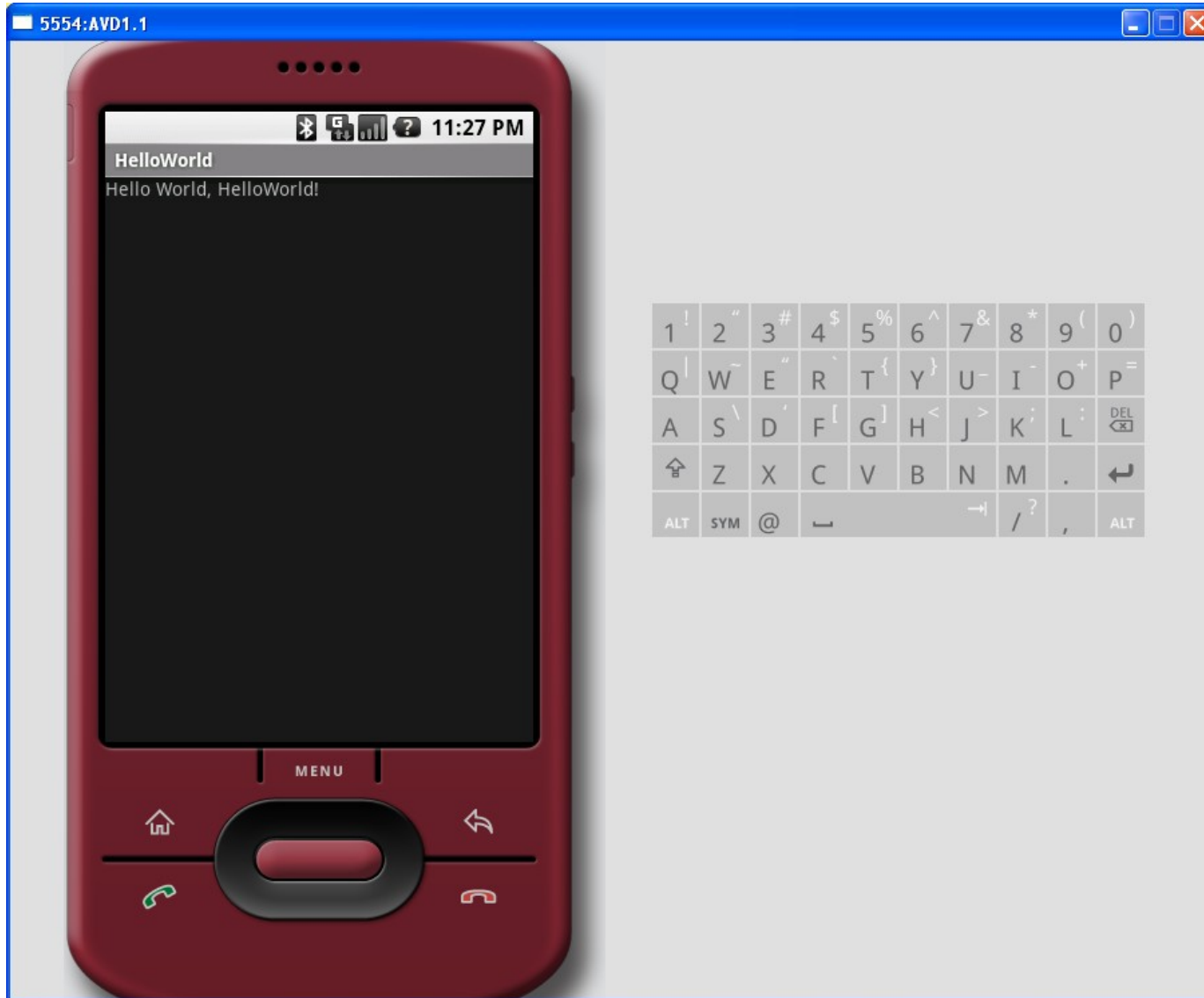
Force create



# Creating Virtual Device



# Run HelloWorld on AVD1.1



# A Quick Look At HelloWorld

```
//HelloWorld.java
package edu.pacificu.cs.HelloWorld;

import android.app.Activity;
import android.os.Bundle;

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

# HelloWorld.java

## Activity

- base class that contains UI components for the application
- similar to a Form for Desktop Apps

## Views

- are the visual components describing the layout of the visual interface

onCreate is overridden to do the activities normal static setup

savedInstanceState is a Bundle that contains the Activities previously frozen state (if one exists)

```
super.onCreate(savedInstanceState) ;
```

setContentView creates a UI using a static View resource in this case

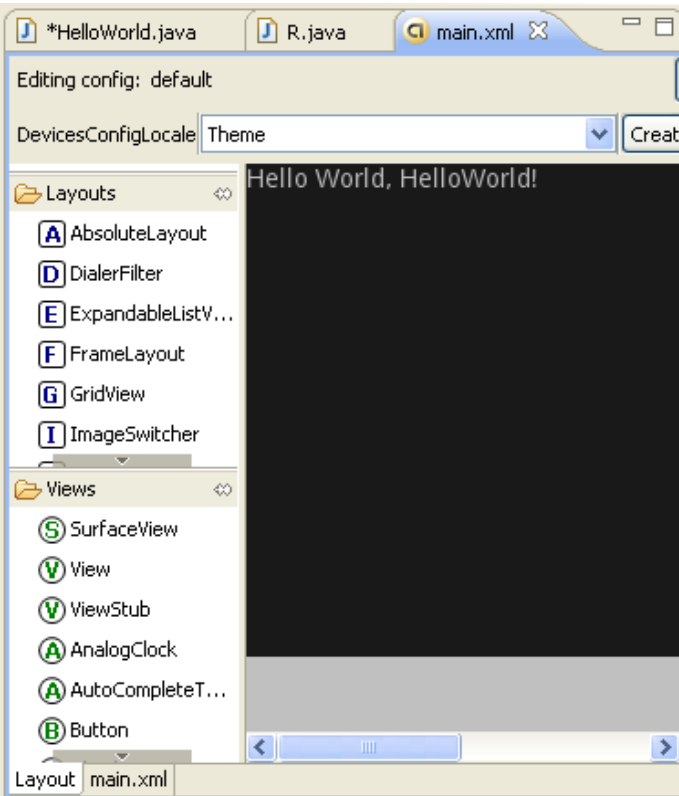
```
setContentView (R.layout.main) ;
```

# Android Resources

Android resources are stored in the res folder which minimally consists of folders:

- drawable
- layout
- values

XML resources are specified in main.xml to describe at UI layout



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```

# Questions

1. What kind of argument does `setContentView` accept?
2. Why is `R.layout.main` an acceptable argument for `setContentView`?
3. Is there a text field in the application? If so, what is the name of the text field?
4. How would you change the program to print out **Hello Yourname**?

# HelloWorld Displayed In Code

## 04.code\HelloWorld

```
package edu.pacificu.cs.HelloWorld;

import android.app.Activity;
import android.os.Bundle;
import android.view.ViewGroup.LayoutParams;
import android.widget.LinearLayout;
import android.widget.TextView;

public class HelloWorld extends Activity
{
    TextView helloTextView; // assigned TextView resource
    private static boolean bUseXML = true;

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

        if (bUseXML)
            useXMLLayout();
        else
            useCodeLayout();
    }
}
```

# HelloWorld Continued

```
// Using XML layout resource to create UI
private void useXMLLayout()
{
    setContentView (R.layout.main);
    helloTextView =
        (TextView) findViewById (R.id.helloTextView);
}
```



# HelloWorld Continued

```
// Create and populate layout in code
```

```
private void useCodeLayout()
```

```
{
```

```
    LinearLayout.LayoutParams layoutParams =
```

```
        new LinearLayout.LayoutParams (LayoutParams.FILL_PARENT,  
                                        LayoutParams.FILL_PARENT);
```

```
    LinearLayout.LayoutParams textViewLayoutParams =
```

```
        new LinearLayout.LayoutParams (LayoutParams.FILL_PARENT,  
                                        LayoutParams.WRAP_CONTENT);
```

```
    LinearLayout linearLayout = new LinearLayout (this);
```

```
    linearLayout.setOrientation (LinearLayout.VERTICAL);
```

```
    helloTextView = new TextView (this);
```

```
    helloTextView.setText ("Hello Yourname");
```

```
    linearLayout.addView (helloTextView, textViewLayoutParams);
```

```
    addContentView (linearLayout, layoutParams);
```

```
}
```

```
}
```

# Android Developers Guide

<http://developer.android.com/guide/index.html>

The screenshot shows the Mozilla Firefox browser displaying the Android Developers Guide website. The browser's address bar shows the URL <http://developer.android.com/guide/index.html>. The website's navigation menu includes links for Home, SDK, Dev Guide (which is highlighted), Reference, Resources, Videos, and Blog. The main content area is titled "The Developer's Guide" and contains the following text:

Welcome to the *Android Dev Guide!* The Dev Guide is a practical introduction to developing applications for Android. It explores the concepts behind Android, the framework for constructing an application, and the tools for developing, testing, and publishing software for the platform.

The Dev Guide holds most of the documentation for the Android platform, except for reference material on the framework API. For API specifications, go to the [Reference](#) tab above.

As you can see in the panel on the left, the Dev Guide is divided into a handful of sections. They are:

- Android Basics**  
An initial orientation to Android — what it is, what it offers, and how your application fits in.
- Framework Topics**  
Discussions of particular parts of the Android framework and API. For an overview of the framework, begin with [Application Fundamentals](#). Then explore other topics — from designing a user interface and setting up resources to storing data and using permissions — as needed.
- Developing**  
Directions for using Android's development and debugging tools, and for testing the results.
- Publishing**  
Instructions on how to prepare your application for deployment and how to publish it when it's ready.
- Best Practices**  
Recommendations on preferred techniques for writing applications that perform efficiently and work well for the user.
- Tutorials and Samples**  
Step-by-step tutorials and sample code demonstrating how an Android application is constructed.
- Appendix**  
Reference information and specifications, as well as FAQs, a glossary

The left sidebar of the website lists various sections under "Android Basics", "Framework Topics", "Developing", "Publishing", "Best Practices", and "Appendix".

# Android Development Tools

- Android Emulator – Android virtual machine
- Dalvik Debug Monitoring Service (DDMS) – used for debugging apps
- Android Asset Packaging Tool (AAPT) – creates distributable Android package files (.apk)
- Android Debug Bridge (ADB) – client-server app providing communication with a running app
- SQLite3 – used to access SQLite database files
- Traceview – used to view trace logs from an Android app
- MkSDCard – creates an SDCard disk image
- dx – converts Java bytecode to Android bytecode
- activityCreator – builds Ant files to compile Android apps without ADT plugin