

Sample code

DT228/3

Dr. Susan McKeever



ANDROID

At a minimum, an app needs..

A layout (i.e. spec for a screen)

An activity – java code to “run” the screen

Manifest file (configuration file)

Default are all created for you in Studio

Project Directory layout

The screenshot displays the Android Studio 2.1.1 interface. The top toolbar includes icons for File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path: MyApplication > app > src > main > java > com > lightcone > myapplication > MainActivity.

The left sidebar contains the Project, Structure, Captures, Favorites, and Recent tabs. The Project tab is active, showing the directory tree for the 'app' module:

- app
 - manifests
 - java
 - com.lightcone.myapplication
 - MainActivity
 - res
 - drawable
 - layout
 - activity_main.xml
 - content_main.xml
 - menu
 - mipmap
 - values
 - Gradle Scripts

The main editor window displays the MainActivity.java file with the following code:

```
package com.lightcone.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

The right sidebar contains the Gradle and Android Model tabs. The Gradle tab is active, showing a green checkmark. The bottom status bar indicates the Gradle build finished in 2s 665ms (4 minutes ago) and the current context is <no context>.

Sample Layout file (res/layout)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Sample Activity class

// doesn't do a whole lot! Just connects to the layout
And displays it

```
package com.example.helloworld;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);
```

```
        // set screen to this layout:
```

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

```
    }
```

```
}
```

Sample Android Manifest layout..

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.tutorialspoint7.myapplication">
```

```
    <application
```

```
        android:allowBackup="true"
```

```
        android:icon="@mipmap/ic_launcher"
```

```
        android:label="@string/app_name"
```

```
        android:supportsRtl="true"
```

```
        android:theme="@style/AppTheme">
```

```
        <activity android:name=".MainActivity">
```

```
            <intent-filter>
```

```
                <action android:name="android.intent.action.MAIN" />
```

```
                <category android:name="android.intent.category.LAUNCHER" />
```

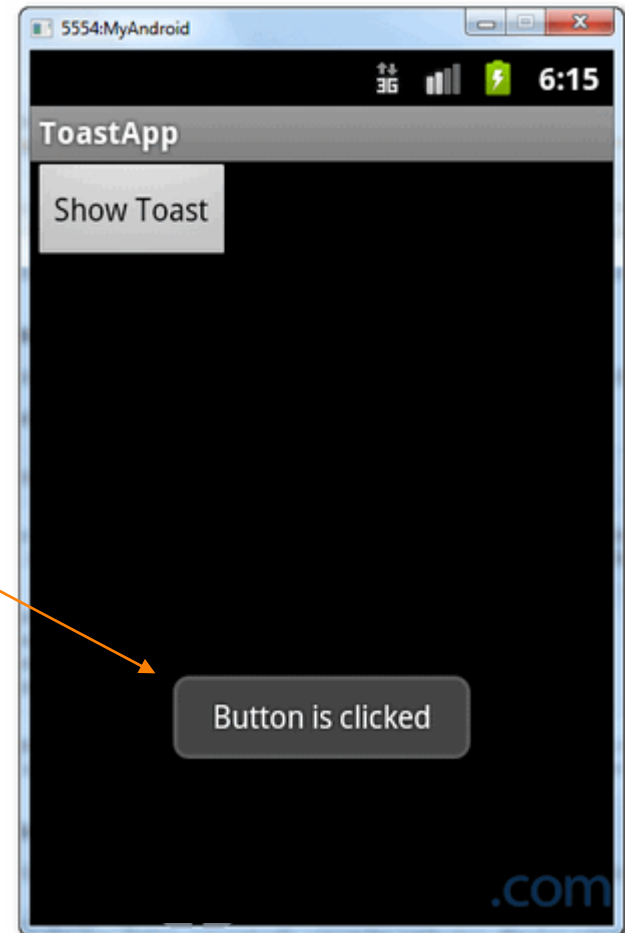
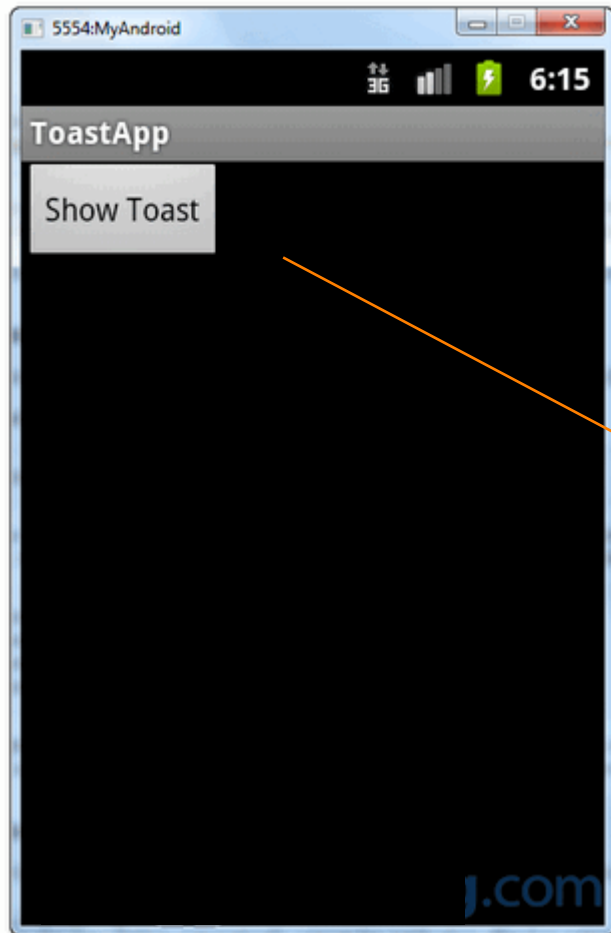
```
            </intent-filter>
```

```
        </activity>
```

```
    </application>
```

```
</manifest>
```

Another example – Clickable screen



XML Layout code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/buttonToast"
        // etc other button attributes
    />

</LinearLayout>
```


Activity class – clickable button

```
public class MainActivity extends Activity {  
    private Button button;  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
  
        button = (Button) findViewById(R.id.buttonToast);  
  
        // anonymous inner class (it instantiates a class that has no name,  
        // and which implements an interface called OnClickListener  
        button.setOnClickListener(new OnClickListener()  
        {  
            public void onClick(View v)  
            {  
                // whatever you want to happen when  
                // button is clicked  
            }  
        });  
    }  
}
```

Note: "anonymous class" code structure.. just one of several ways to set up