Practical: 1

Practical: Design Login activity and implement control events: Use EditText, Checkbox and Buttons

activity_main.xml

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout

            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="40dp"
            android:layout_marginTop="20dp"
            android:layout_marginRight="40dp"
            android:layout_marginBottom="15dp"
            android:orientation="vertical">


            <TextView android:id="@+id/login"
                android:layout_width="match_parent"
                android:layout_height="wrap_content
                "
                android:layout_marginBottom="30dp"
                android:fontFamily="cursive"
                android:gravity="center"
                android:text="Login"
                android:textSize="50sp" />

            <EditText
                android:id="@+id/editTextTextEmailAddress"
                android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp" android:ems="10"
    android:gravity="center" android:hint="gmail"
    android:inputType="textEmailAddress"
    android:padding="9dp" />

<EditText android:id="@+id/editTextTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_marginBottom="20dp"
    android:ems="10" android:gravity="center"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="9dp" />

<TextView
    android:id="@+id/languagetesxid"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Known Langage: "
    android:textSize="20dp" />

<CheckBox
    android:id="@+id/javaCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="JAVA"
    android:textSize="18sp"
    android:textStyle="italic" />

<CheckBox
    android:id="@+id/androidCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Android"
    android:textSize="18sp"
    android:textStyle="italic" />
```

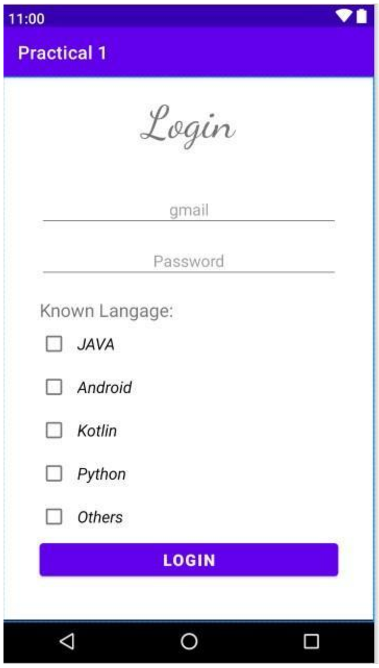
```
        <CheckBox
            android:id="@+id/kotlinCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="10dp"
            android:text="Kotlin"
            android:textSize="18sp"
            android:textStyle="italic" />

        <CheckBox
            android:id="@+id/pythonCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="10dp"
            android:text="Python"
            android:textSize="18sp"
            android:textStyle="italic" />

        <CheckBox
            android:id="@+id/otherCheckBox"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:checked="false"
            android:padding="10dp"
            android:text="Others"
            android:textSize="18sp"
            android:textStyle="italic" />

        <Button android:id="@+id/loginnbtn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content
            " android:text="Login"
            android:textColor="#FBF5F5"
            android:textSize="18sp"
            android:textStyle="bold" />

    </LinearLayout>
</LinearLayout>
```

```
</LinearLayout>
```

MainActivtity.java

```
package com.example.practical1;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; public class MainActivity
extends AppCompatActivity {

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

Output:



Practical: 2

Practical: Implement Practical 1 using following layouts:
1. Linear Layout
2. Relative Layout
3. Table Layout

1) Linear Layout:

activity_main.xml

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

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout

            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_marginLeft="40dp"
            android:layout_marginTop="20dp"
            android:layout_marginRight="40dp"
            android:layout_marginBottom="15dp"
            android:orientation="vertical">


            <TextView android:id="@+id/login"
                android:layout_width="match_parent"
                android:layout_height="wrap_content
                "
                android:layout_marginBottom="30dp"
                android:fontFamily="cursive"
                android:gravity="center"
                android:text="Login"
                android:textSize="50sp" />
```

```
<EditText
    android:id="@+id/editTextTextEmailAddress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:ems="10" android:gravity="center"
    android:hint="gmail"
    android:inputType="textEmailAddress"
    android:padding="9dp" />

<EditText android:id="@+id/editTextTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_marginBottom="20dp"
    android:ems="10" android:gravity="center"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="9dp" />

<TextView
    android:id="@+id/languagetesxid"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Known Langage: "
    android:textSize="20dp" />

<CheckBox
    android:id="@+id/javaCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="JAVA"
    android:textSize="18sp"
    android:textStyle="italic" />

<CheckBox
    android:id="@+id/androidCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Android"
    android:textSize="18sp"
    android:textStyle="italic" />

<CheckBox
    android:id="@+id/kotlinCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Kotlin"
    android:textSize="18sp"
    android:textStyle="italic" />

<CheckBox
    android:id="@+id/pythonCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Python"
    android:textSize="18sp"
    android:textStyle="italic" />

<CheckBox
    android:id="@+id/otherCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Others"
    android:textSize="18sp"
    android:textStyle="italic" />

<Button android:id="@+id/loginnbtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:text="Login"
    android:textColor="#FBF5F5"
    android:textSize="18sp"
    android:textStyle="bold" />
```

```
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

2) Relative Layout:

activity_relative_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    tools:context=".RelativeLayout">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="40dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="40dp"
        android:layout_marginBottom="15dp">

        <TextView android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:fontFamily="cursive"
            android:gravity="center"
            android:text="Login"
            android:textSize="50dp" />

        <EditText
            android:id="@+id/editTextTextEmailAddress"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@+id/textView"
            android:layout_marginTop="14dp" android:ems="10"
            android:gravity="center" android:hint="gmail"
            android:inputType="textEmailAddress"
            android:padding="9dp" />
```

```
<EditText android:id="@+id/editTextTextPassword"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_below="@+id/editTextTextEmailAddress
   " android:layout_marginTop="15dp"
   android:layout_marginBottom="20dp" android:ems="10"
   android:gravity="center" android:hint="Password"
   android:inputType="textPassword"
   android:padding="9dp" />

<TextView
   android:id="@+id/languagetesxid"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_below="@+id/editTextTex
   tPassword" android:text="Known
   Langage: " android:textSize="20dp" />

<CheckBox android:id="@+id/javaCheckBox"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_below="@+id/languagetesxid"
   android:checked="false" android:padding="10dp"
   android:text="JAVA" android:textSize="18sp"
   android:textStyle="italic" />

<CheckBox android:id="@+id/androidCheckBox"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_below="@+id/javaCheckBox"
   android:checked="false" android:padding="10dp"
   android:text="Android" android:textSize="18sp"
   android:textStyle="italic" />

<CheckBox android:id="@+id/kotlinCheckBox"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_below="@+id/androidCheckBox"
   android:checked="false" android:padding="10dp"
   android:text="Kotlin" android:textSize="18sp"
   android:textStyle="italic" />

<CheckBox android:id="@+id/pythonCheckBox"
   android:layout_width="wrap_content"
```

```
         android:layout_height="wrap_content"
         android:layout_below="@+id/kotlinCheckBox"
         android:checked="false" android:padding="10dp"
         android:text="Python"
         android:textSize="18sp"
         android:textStyle="italic" />

      <CheckBox android:id="@+id/otherCheckBox"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_below="@+id/pythonCheckBox"
         android:checked="false" android:padding="10dp"
         android:text="Others" android:textSize="18sp"
         android:textStyle="italic" />

      <Button android:id="@+id/loginnbtn"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_below="@+id/otherCheckBox
         " android:layout_marginTop="9dp"
         android:text="Login"
         android:textColor="#FBF5F5"
         android:textSize="18sp" android:textStyle="bold"
         />

    </RelativeLayout>
 </RelativeLayout>
```

3) Table Layout:

activity_table_layout.xml

```
 <?xml version="1.0" encoding="utf-8"?>
 <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
     xmlns:app="http://schemas.android.com/apk/res-auto"
     xmlns:tools="http://schemas.android.com/tools"
     android:layout_width="match_parent"
     android:layout_height="match_parent" android:layout_marginLeft="40dp"
     android:layout_marginTop="20dp" android:layout_marginRight="40dp"
     android:layout_marginBottom="15dp"
     tools:context=".TableLayout">
    <TableRow
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center">
    <TextView android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content
        " android:fontFamily="cursive"
        android:gravity="center"
        android:text="Login"
        android:textSize="50sp" /> </TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">
    <EditText
        android:id="@+id/editTextTextEmailAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:ems="10" android:gravity="center"
        android:hint="gmail"
        android:inputType="textEmailAddress"
        android:padding="9dp" />
</TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">
    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:hint="Password"
        android:inputType="textPassword" />
</TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
android:id="@+id/languagetesxid"
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:text="Known Langage: "
android:textSize="20dp" /> </TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
   <CheckBox
       android:id="@+id/javaCheckBox"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_centerHorizontal="true"
       android:checked="false"
       android:padding="10dp"
       android:text="JAVA"
       android:textSize="18sp"
       android:textStyle="italic" />
</TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
   <CheckBox
       android:id="@+id/androidCheckBox"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_centerHorizontal="true"
       android:checked="false"
       android:padding="10dp"
       android:text="Android"
       android:textSize="18sp"
       android:textStyle="italic" /> </TableRow>
<TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
   <CheckBox
       android:id="@+id/kotlinCheckBox"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_centerHorizontal="true"
       android:checked="false"
       android:padding="10dp"
       android:text="Kotlin"
       android:textSize="18sp"
       android:textStyle="italic" /> </TableRow>
```

```
  <TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <CheckBox
    android:id="@+id/pythonCheckBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:checked="false"
    android:padding="10dp"
    android:text="Python"
    android:textSize="18sp"
      android:textStyle="italic" /> </TableRow>
  <TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <CheckBox
      android:id="@+id/otherCheckBox"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:checked="false"
      android:padding="10dp"
      android:text="Others"
      android:textSize="18sp"
      android:textStyle="italic" /> </TableRow>

  <TableRow
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">

    <Button android:id="@+id/loginnbtn"
      android:layout_width="match_parent"
      android:layout_height="wrap_content
      " android:text="Login"
      android:textColor="#FBF5F5"
      android:textSize="18sp"
      android:textStyle="bold" />
  </TableRow>

</TableLayout>
```

MainActivtity.java

```
package com.example.practical1;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle; public class MainActivity
extends AppCompatActivity {

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

Output:

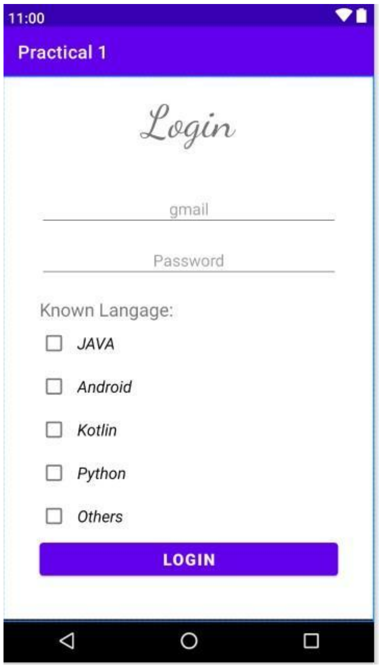


Figure 1 [Linear Layout]

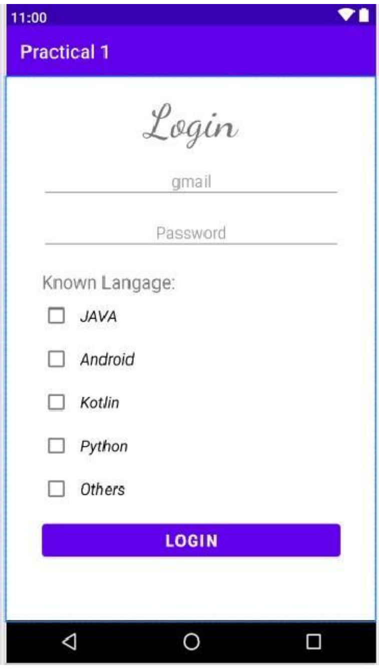


figure 2 [Relative Layout]

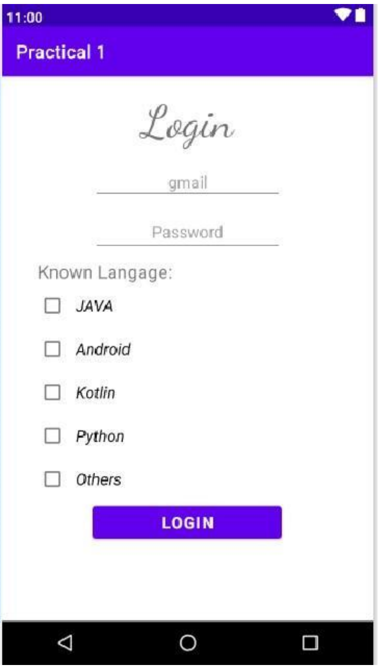


figure 3 [Linear Layout]

Practical: 3

Practical: Create Activities & implement following
1. Implicit intent
2. Explicit Intent
3. StartActivityForResult

Implicit Intent

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
android:orientation="vertical"     android:padding="16dp"
    tools:context="com.example.android.implicitintents.MainActivity">

    <EditText
        android:id="@+id/website_edittext"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/edittext_uri"/>
     <Button
        android:id="@+id/open_website_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="24dp"        android:text="@string/button_uri"
android:onClick="openWebsite"/>

    <EditText
        android:id="@+id/location_edittext"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/edittext_loc"/>
     <Button
        android:id="@+id/open_location_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="24dp"        android:text="@string/button_loc"
android:onClick="openLocation"/>

    <EditText
```

```
        android:id="@+id/share_edittext"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/edittext_share"/>

    <Button
        android:id="@+id/share_text_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="24dp"
android:text="@string/button_share"        android:onClick="shareText"/>

</LinearLayout>
```

ActivityMain.java

```
package com.example.android.implicitintents;
 import android.content.Intent;
import android.net.Uri; import
android.os.Bundle;
import android.support.v4.app.ShareCompat; import
android.support.v7.app.AppCompatActivity; import
android.util.Log; import android.view.View; import
android.widget.EditText;

/**
 * The ImplicitIntents app contains three buttons for sending implicit intents:
 * - Open a URL in a browser
 * - Find a location on a map
 * - Share a text string
 */
public class MainActivity extends AppCompatActivity {
     private EditText mWebsiteEditText;
private EditText mLocationEditText;
private EditText mShareTextEditText;

    /**
 * Initializes the activity.
     *
 * @param savedInstanceState The current state data.
     */     @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_main);
```

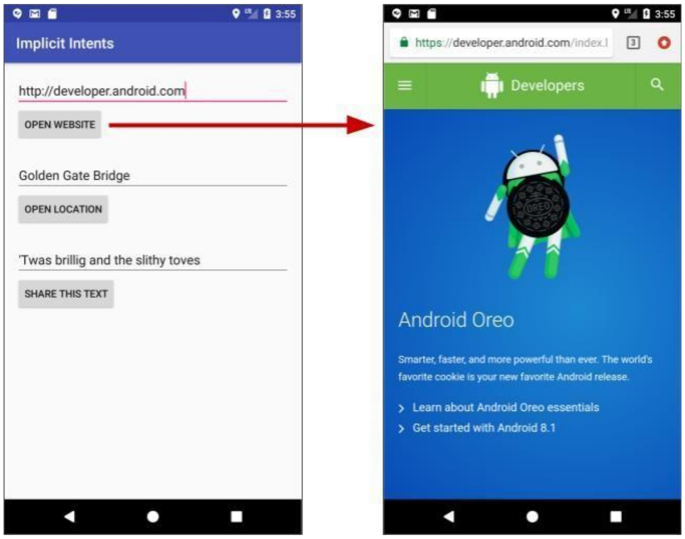
```
     mWebsiteEditText = findViewById(R.id.website_edittext);
mLocationEditText = findViewById(R.id.location_edittext);
mShareTextEditText = findViewById(R.id.share_edittext);
  }

  /**
* Handles the onClick for the "Open Website" button. Gets the URI
* from the edit text and sends an implicit intent for that URL.
   *
* @param view The view (Button) that was clicked.
   */
  public void openWebsite(View view) {
    // Get the URL text.
    String url = mWebsiteEditText.getText().toString();

    // Parse the URI and create the intent.
    Uri webpage = Uri.parse(url);
    Intent intent = new Intent(Intent.ACTION_VIEW, webpage);

    // Find an activity to hand the intent and start that activity.
    if (intent.resolveActivity(getPackageManager()) != null) {          startActivity(intent);
    } else {
      Log.d("ImplicitIntents", "Can't handle this!");
    }
  }

  /**
* Handles the onClick for the "Open Location" button. Gets the location      * text from
  the edit text and sends an implicit intent for that location.
   *
* The location text can be any searchable geographic location.
   *
* @param view The view (Button) that was clicked.
   */
  public void openLocation(View view) {
    // Get the string indicating a location. Input is not validated; it is
    // passed to the location handler intact.
    String loc = mLocationEditText.getText().toString();

    // Parse the location and create the intent.
    Uri addressUri = Uri.parse("geo:0,0?q=" + loc);
    Intent intent = new Intent(Intent.ACTION_VIEW, addressUri);

    // Find an activity to handle the intent, and start that activity.
```

```
        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivity(intent);
        } else {
            Log.d("ImplicitIntents", "Can't handle this intent!");
        }
    }

    /**
 * Handles the onClick for the "Share This Text" button. The
 * implicit intent here is created by the  {@link ShareCompat.IntentBuilder}       * class.
    An app chooser appears with the available options for sharing.
     *
 * ShareCompat.IntentBuilder is from the v4 Support Library.
     *
 * @param view The view (Button) that was clicked.
     */
    public void shareText(View view) {
        String txt = mShareTextEditText.getText().toString();
        String mimeType = "text/plain";
        ShareCompat.IntentBuilder
                .from(this)
                .setType(mimeType)
                .setChooserTitle(R.string.share_text_with)
                .setText(txt)
                .startChooser();
    } }
```

strings.xml

```
<resources>
    <string name="app_name">Implicit Intents</string>

    <string name="edittext_uri">http://developer.android.com</string>
    <string name="button_uri">Open Website</string>

    <string name="edittext_loc">Golden Gate Bridge</string>
    <string name="button_loc">Open Location</string>

    <string name="edittext_share">\'Twas brillig and the slithy toves</string>
    <string name="button_share">Share This Text</string>
    <string name="share_text_with">Share this text with:</string>

</resources>
```

Output:

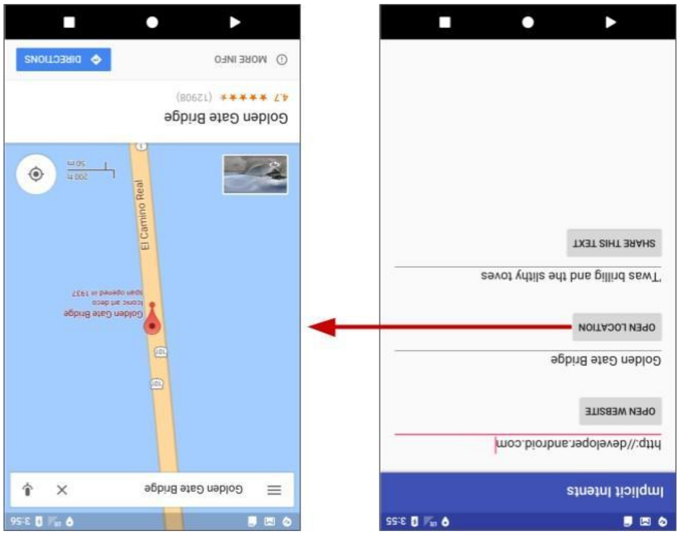


3:55
Implicit Intents
OPEN WEBSITE
OPEN LOCATION
SHARE THIS TEXT
https://developer.android.com/index.l
Developers
Android Oreo
Smarter, faster, and more powerful than ever. The world's favorite cookie is your new favorite Android release.
Learn about Android Oreo essentials
Get started with Android 8.1

http://developer.android.com

Golden Gate Bridge

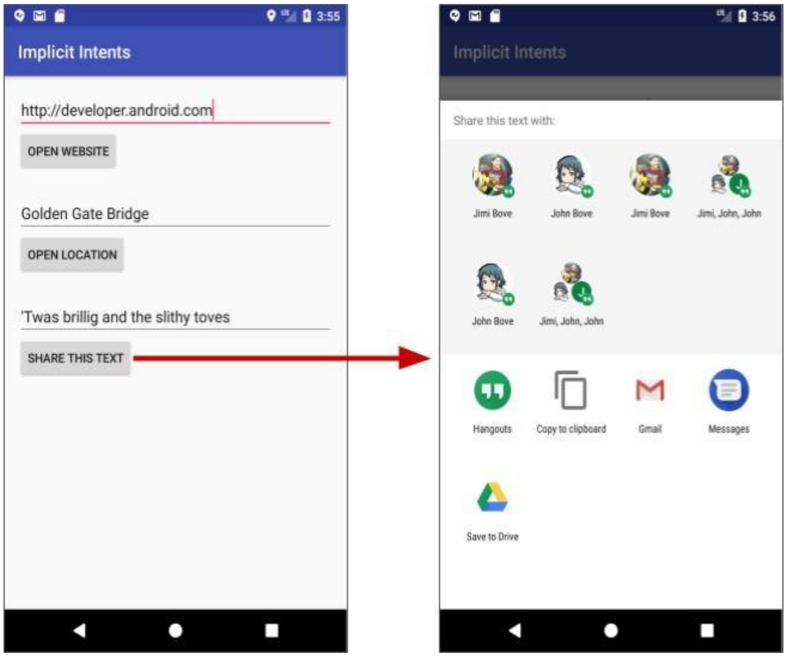
'Twas brillig and the slithy toves



http://developer.android.com

Golden Gate Bridge

'Twas brillig and the slithy toves



http://developer.android.com

Golden Gate Bridge

'Twas brillig and the slithy toves

Implicit Intents Receiver activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
   tools:context="com.example.android.implicitintentsreceiver.MainActivity">
   <TextView
       android:id="@+id/text_uri_message"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textSize="18sp"        android:textStyle="bold"
       app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package com.example.android.implicitintentsreceiver;
 import android.content.Intent;
import android.net.Uri; import
android.os.Bundle;
import android.support.v7.app.AppCompatActivity; import
android.widget.TextView;

/**
* The ImplicitIntentsReceiver app registers itself for implicit intents  * that come from
  browsable links (URLs) with the scheme:http and
* host:developer.android.com (see AndroidManifest.xml).
 *
* If it receives that intent, the app just prints the incoming URI
* to a TextView.
 *
* ImplicitIntentsReceiver is intended to be used in conjunction with
* the ImplicitIntents app, but will receive a matching implicit intent
* from any app (for example, a link in an email.)
 */
public class MainActivity extends AppCompatActivity {

  /**
* Initializes the activity.
   *
* @param savedInstanceState The current state data
   */    @Override
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_main);

     Intent intent = getIntent();        Uri uri
= intent.getData();        if (uri != null) {
        String uri_string = getString(R.string.uri_label)
             + uri.toString();
        TextView textView = findViewById(R.id.text_uri_message);
        textView.setText(uri_string);
     }
  } }
```

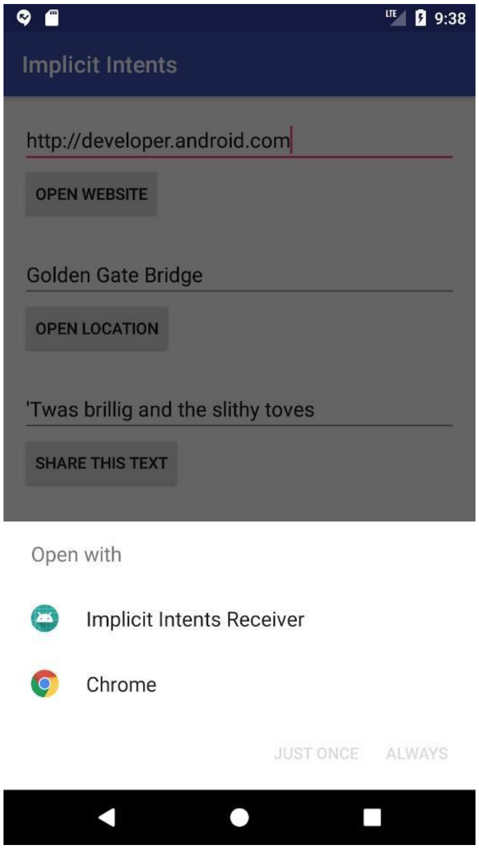
strings.xml

```
<resources>
   <string name="app_name">Implicit Intents Receiver</string>
   <string name="uri_label">URI:</string>
</resources>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.android.implicitintentsreceiver">

   <application
      android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
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>

         <intent-filter>
            <action android:name="android.intent.action.VIEW" />
            <category android:name="android.intent.category.DEFAULT" />
            <category android:name="android.intent.category.BROWSABLE" />             <data
android:scheme="http" android:host="developer.android.com" />
         </intent-filter>
      </activity>
   </application>

</manifest>
```

Output:



http://developer.android.com

Golden Gate Bridge

'Twas brillig and the slithy toves

Explicit Intent & StartActivityForResult

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  ~ Copyright (C) 2018 Google Inc.
  ~
  ~ Licensed under the Apache License, Version 2.0 (the "License");   ~ you
may not use this file except in compliance with the License.
  ~ You may obtain a copy of the License at
  ~
  ~     http://www.apache.org/licenses/LICENSE-2.0
  ~
  ~ Unless required by applicable law or agreed to in writing, software
  ~ distributed under the License is distributed on an "AS IS" BASIS,   ~ WITHOUT
WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.   ~ See
the License for the specific language governing permissions and
  ~ limitations under the License.
  -->

<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
    tools:context="com.example.android.twoactivities.MainActivity">

    <TextView
        android:id="@+id/text_header_reply"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"        android:layout_marginTop="16dp"
android:text="@string/text_header_reply"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textStyle="bold"        android:visibility="invisible"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/text_message_reply"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"        android:layout_marginLeft="8dp"
android:layout_marginTop="8dp"        android:visibility="invisible"
app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_header_reply" />

    <Button
        android:id="@+id/button_main"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginRight="16dp"
android:text="@string/button_main"
android:onClick="launchSecondActivity"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent" />

    <EditText
        android:id="@+id/editText_main"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
```

```
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"        android:ems="10"
        android:hint="@string/editText_main"
android:inputType="textLongMessage"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/button_main"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package com.example.android.twoactivities;
 import android.content.Intent; import
android.os.Bundle; import
android.support.v7.app.AppCompatActivity; import
android.util.Log; import android.view.View; import
android.widget.EditText; import
android.widget.TextView;

/**
 * The TwoActivities app contains two activities and sends messages  *
(intents) between them.
 */
public class MainActivity extends AppCompatActivity {
    // Class name for Log tag
    private static final String LOG_TAG = MainActivity.class.getSimpleName();
    // Unique tag required for the intent extra
    public static final String EXTRA_MESSAGE
            = "com.example.android.twoactivities.extra.MESSAGE";
    // Unique tag for the intent reply     public static
final int TEXT_REQUEST = 1;

    // EditText view for the message     private
EditText mMessageEditText;     // TextView for
the reply header     private TextView
mReplyHeadTextView;     // TextView for the
reply body     private TextView
mReplyTextView;

    /**
*  Initializes the activity.
     *
*  @param savedInstanceState The current state data.
```

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

       // Initialize all the view variables.         mMessageEditText =
findViewById(R.id.editText_main);         mReplyHeadTextView =
findViewById(R.id.text_header_reply);         mReplyTextView =
findViewById(R.id.text_message_reply);
   }

   /**
* Handles the onClick for the "Send" button. Gets the value of the main
EditText,
* creates an intent, and launches the second activity with that intent.
    *
* The return intent from the second activity is onActivityResult().
    *
* @param view The view (Button) that was clicked.
    */
   public void launchSecondActivity(View view) {
       Log.d(LOG_TAG, "Button clicked!");
       Intent intent = new Intent(this, SecondActivity.class);         String message
= mMessageEditText.getText().toString();
intent.putExtra(EXTRA_MESSAGE, message);
startActivityForResult(intent, TEXT_REQUEST);
   }

   /**
* Handles the data in the return intent from SecondActivity.
    *
* @param requestCode Code for the SecondActivity request.
* @param resultCode Code that comes back from SecondActivity.
* @param data Intent data sent back from SecondActivity.
    */     @Override
   public void onActivityResult(int requestCode, int resultCode, Intent data) {
super.onActivityResult(requestCode, resultCode, data);

       // Test for the right intent reply.         if
(requestCode == TEXT_REQUEST) {
           // Test to make sure the intent reply result was good.             if
(resultCode == RESULT_OK) {
               String reply = data.getStringExtra(SecondActivity.EXTRA_REPLY);
               // Make the reply head visible.
               mReplyHeadTextView.setVisibility(View.VISIBLE);
```

```
                // Set the reply and make it visible.
                mReplyTextView.setText(reply);
                mReplyTextView.setVisibility(View.VISIBLE);
            }
        }
    }
}
```

activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"         android:layout_width="wrap_content"
android:layout_height="wrap_content"         android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"         android:layout_marginTop="16dp"
android:text="@string/text_header"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/text_message"
        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"         android:layout_marginLeft="8dp"
android:layout_marginTop="8dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/text_header" />
     <Button
        android:id="@+id/button_second"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginRight="16dp"
android:text="@string/button_second"         android:onClick="returnReply"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent" />

   <EditText
       android:id="@+id/editText_second"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"       android:ems="10"
       android:hint="@string/editText_second"
android:inputType="textLongMessage"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/button_second"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

SecondActivity.java

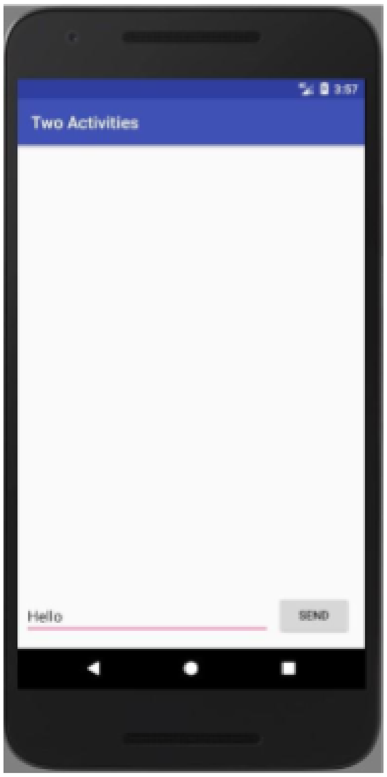
```
package com.example.android.twoactivities;
 import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.EditText; import
android.widget.TextView;

/**
* SecondActivity defines the second activity in the app. It is  * launched from an intent
   with a message, and sends an intent  * back with a second message.
 */ public class SecondActivity extends AppCompatActivity {
   // Unique tag for the intent reply.
   public static final String EXTRA_REPLY =
          "com.example.android.twoactivities.extra.REPLY";

   // EditText for the reply.     private
EditText mReply;

   /**
* Initializes the activity.
    *
* @param savedInstanceState The current state data
    */    @Override
```

```
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_second);

    // Initialize view variables.        mReply =
findViewById(R.id.editText_second);

    // Get the intent that launched this activity, and the message in
    // the intent extra.
    Intent intent = getIntent();
    String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

    // Put that message into the text_message TextView        TextView
textView = findViewById(R.id.text_message);
textView.setText(message);
  }

  /**
* Handles the onClick for the "Reply" button. Gets the message from the      * second
  EditText, creates an intent, and returns that message back to
* the main activity.
   *
* @param view The view (Button) that was clicked.
   */
  public void returnReply(View view) {
    // Get the reply message from the edit text.
    String reply = mReply.getText().toString();

    // Create a new intent for the reply, add the reply message to it
    // as an extra, set the intent result, and close the activity.        Intent
replyIntent = new Intent();        replyIntent.putExtra(EXTRA_REPLY, reply);
setResult(RESULT_OK, replyIntent);        finish();
  } }
```

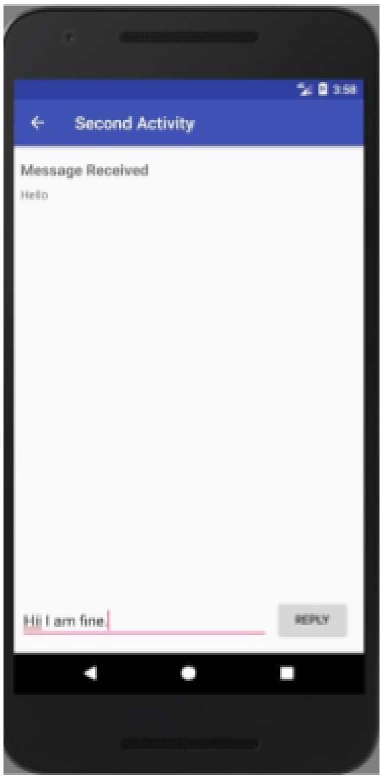
strings.xml

```
<resources>
  <string name="app_name">Two Activities</string>
  <string name="button_main">Send</string>
  <string name="editText_main">Enter Your Message Here</string>
  <string name="text_header">Message Received</string>
  <string name="button_second">Reply</string>
  <string name="editText_second">Enter Your Reply Here</string>
  <string name="text_header_reply">Reply Received</string>
</resources>
```

Output:



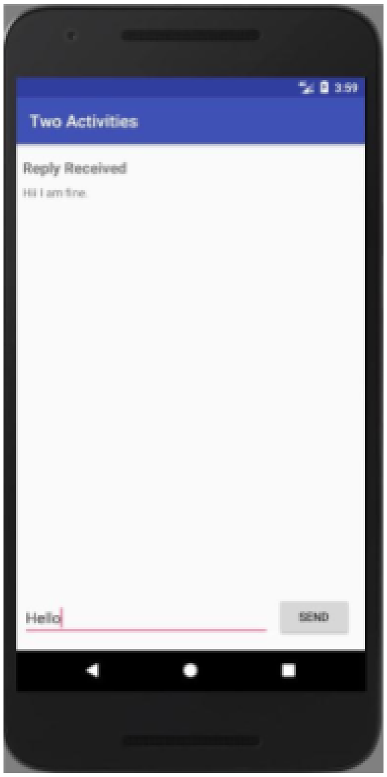
Hello



Message Received

Hello

Hi! I am fine.



Reply Received

Hi! I am fine.

Hello

Practical: 4

Practical: Implement activity Lifecycle and State Callbacks

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
   tools:context="com.example.android.twoactivities.MainActivity">

   <TextView
      android:id="@+id/text_header_reply"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"        android:layout_marginTop="16dp"
android:text="@string/text_header_reply"
      android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textStyle="bold"        android:visibility="invisible"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

   <TextView
      android:id="@+id/text_message_reply"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"        android:layout_marginLeft="8dp"
android:layout_marginTop="8dp"        android:visibility="invisible"
app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/text_header_reply" />
    <Button
      android:id="@+id/button_main"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginRight="16dp"
android:text="@string/button_main"
android:onClick="launchSecondActivity"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent" />



<EditText
```

```
        android:id="@+id/editText_main"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"         android:ems="10"
        android:hint="@string/editText_main"
android:inputType="textLongMessage"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/button_main"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package com.example.android.twoactivities;
 import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log; import android.view.View;
import android.widget.EditText; import
android.widget.TextView;

/**
 * Version of TwoActivities app that prints messages to the logs   * on
Activity lifecycle state changes.
 */
public class MainActivity extends AppCompatActivity {
    // Class name for Log tag
    private static final String LOG_TAG = MainActivity.class.getSimpleName();
    // Unique tag required for the intent extra      public static
final String EXTRA_MESSAGE
            = "com.example.android.twoactivities.extra.MESSAGE";
    // Unique tag for the intent reply      public static
final int TEXT_REQUEST = 1;

    // EditText view for the message      private
EditText mMessageEditText;      // TextView for
the reply header      private TextView
mReplyHeadTextView;      // TextView for the
reply body      private TextView
mReplyTextView;

    @Override
```

```
   protected void onStart() {
super.onStart();        Log.d(LOG_TAG,
"onStart");     }

   /**
* Maintains the Activity state across configuration changes.
    *
* @param outState Activity state data to save
    */     @Override
   public void onSaveInstanceState(Bundle outState) {
super.onSaveInstanceState(outState);
       if (mReplyHeadTextView.getVisibility() == View.VISIBLE) {
outState.putBoolean("reply_visible", true);
           outState.putString("reply_text", mReplyTextView.getText().toString());
       }
   }

   /**
* Initializes the activity.
    *
* @param savedInstanceState The current state data
    */     @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_main);

       // Log the start of the onCreate() method.
       Log.d(LOG_TAG, "-------");
       Log.d(LOG_TAG, "onCreate");

       // Initialize all the view variables.        mMessageEditText =
findViewById(R.id.editText_main);        mReplyHeadTextView =
findViewById(R.id.text_header_reply);        mReplyTextView =
findViewById(R.id.text_message_reply);

       // Restore the state.
       if (savedInstanceState != null) {            boolean
isVisible = savedInstanceState
.getBoolean("reply_visible");            if (isVisible) {
               mReplyHeadTextView.setVisibility(View.VISIBLE);
mReplyTextView.setText(savedInstanceState                     .getString("reply_text"));
mReplyTextView.setVisibility(View.VISIBLE);
           }
       }
   }
   /**
*  Handles the onClick for the "Send" button. Gets the value of the main       * EditText,
   creates an intent, and launches the second activity with
*  that intent.
    *
*  The return intent from the second activity is onActivityResult().
    *
*  @param view The view (Button) that was clicked.
    */
   public void launchSecondActivity(View view) {
       Log.d(LOG_TAG, "Button clicked!");
       Intent intent = new Intent(this, SecondActivity.class);        String message
= mMessageEditText.getText().toString();
intent.putExtra(EXTRA_MESSAGE, message);
startActivityForResult(intent, TEXT_REQUEST);
   }

   /**
*  Handles the data in the return intent from SecondActivity.
    *
*  @param requestCode Code for the SecondActivity request.
*  @param resultCode Code that comes back from SecondActivity.
*  @param data Intent data sent back from SecondActivity.
    */    @Override
   public void onActivityResult(int requestCode, int resultCode, Intent data) {
super.onActivityResult(requestCode, resultCode, data);

       // Test for the right intent reply.         if
(requestCode == TEXT_REQUEST) {
           // Test to make sure the intent reply result was good.             if
(resultCode == RESULT_OK) {
               String reply = data.getStringExtra(SecondActivity.EXTRA_REPLY);
               // Make the reply head visible.
               mReplyHeadTextView.setVisibility(View.VISIBLE);

               // Set the reply and make it visible.
               mReplyTextView.setText(reply);
               mReplyTextView.setVisibility(View.VISIBLE);
           }
       }
   }

   @Override
```

```
    protected void onPause() {        super.onPause();
        Log.d(LOG_TAG, "onPause");
    }

    @Override
    protected void onRestart() {        super.onRestart();
        Log.d(LOG_TAG, "onRestart");
    }

    @Override
    protected void onResume() {        super.onResume();
        Log.d(LOG_TAG, "onResume");
    }

    @Override     protected void
onStop() {        super.onStop();
        Log.d(LOG_TAG, "onStop");
    }

    @Override
    protected void onDestroy() {        super.onDestroy();
        Log.d(LOG_TAG, "onDestroy");
    }
}
```

activity_second.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
    tools:context="com.example.android.twoactivities.SecondActivity">

    <TextView
        android:id="@+id/text_header"        android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginStart="8dp"
android:layout_marginLeft="8dp"        android:layout_marginTop="16dp"
android:text="@string/text_header"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
   <TextView
       android:id="@+id/text_message"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="8dp"        android:layout_marginLeft="8dp"
android:layout_marginTop="8dp"
app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toBottomOf="@+id/text_header" />
    <Button
       android:id="@+id/button_second"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginRight="16dp"
android:text="@string/button_second"        android:onClick="returnReply"
       app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent" />

   <EditText
       android:id="@+id/editText_second"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="16dp"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"        android:ems="10"
       android:hint="@string/editText_second"
android:inputType="textLongMessage"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/button_second"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
```

SecondActivity.java

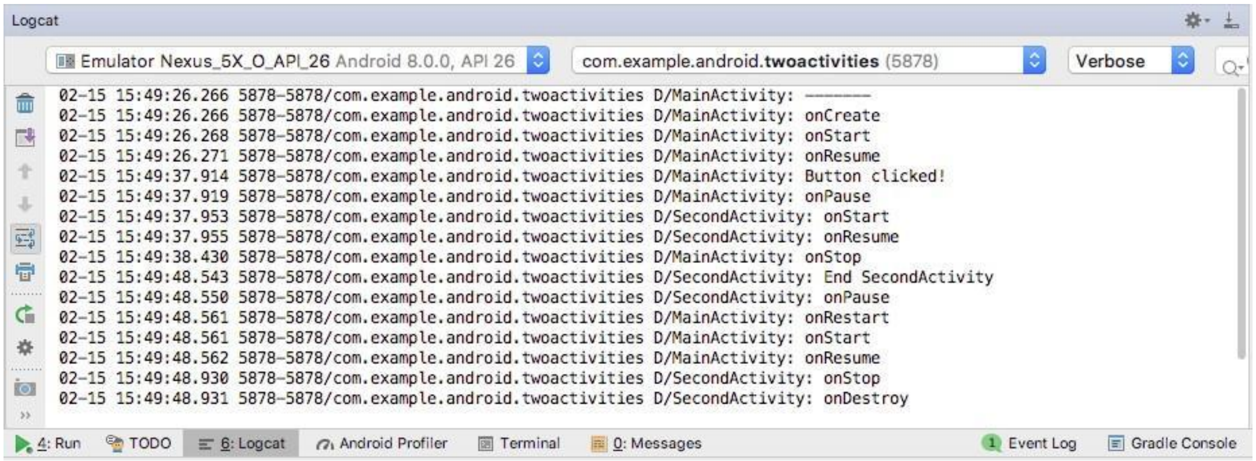
```
package com.example.android.twoactivities;

import android.content.Intent; import
android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log; import android.view.View;
import android.widget.EditText; import
android.widget.TextView;

/**
```

```
* SecondActivity defines the second activity in the app. It is launched
* from an intent with a message, and sends an intent back with a second  * message.
  */
public class SecondActivity extends AppCompatActivity {
  // Class name for Log tag.     private static
final String LOG_TAG
         = SecondActivity.class.getSimpleName();
  // Unique tag for the intent reply.
  public static final String EXTRA_REPLY =
         "com.example.android.twoactivities.extra.REPLY";

  // EditText for the reply.     private
EditText mReply;

  @Override
  protected void onStart() {
super.onStart();        Log.d(LOG_TAG,
"onStart");
  }

  /**
* Initializes the activity.
   *
* @param savedInstanceState The current state data.
   */     @Override
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_second);

     // Initialize view variables.         mReply =
findViewById(R.id.editText_second);

     // Get the intent that launched this activity, and the message in
     // the intent extra.
     Intent intent = getIntent();
     String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

     // Put that message into the text_message TextView.         TextView
textView = findViewById(R.id.text_message);
textView.setText(message);
  }

  /**
* Handles the onClick for the "Reply" button. Gets the message from the      * second
  EditText, creates an intent, and returns that message back to
* the main activity.
     *
* @param view The view (Button) that was clicked.
     */
    public void returnReply(View view) {
        // Get the reply message from the edit text.
        String reply = mReply.getText().toString();

        // Create a new intent for the reply, add the reply message to it         // as an extra,
set the intent result, and close the activity.
        Intent replyIntent = new Intent();
replyIntent.putExtra(EXTRA_REPLY, reply);
setResult(RESULT_OK, replyIntent);         Log.d(LOG_TAG,
"End SecondActivity");         finish();
    }

    @Override     protected void
onPause() {         super.onPause();
Log.d(LOG_TAG, "onPause");
    }

    @Override
    protected void onRestart() {
super.onRestart();         Log.d(LOG_TAG,
"onRestart");
    }

    @Override     protected void
onResume() {         super.onResume();
Log.d(LOG_TAG, "onResume");
    }

    @Override     protected void
onStop() {         super.onStop();
Log.d(LOG_TAG, "onStop");
    }

    @Override
    protected void onDestroy() {
super.onDestroy();         Log.d(LOG_TAG,
"onDestroy");
    } }
```

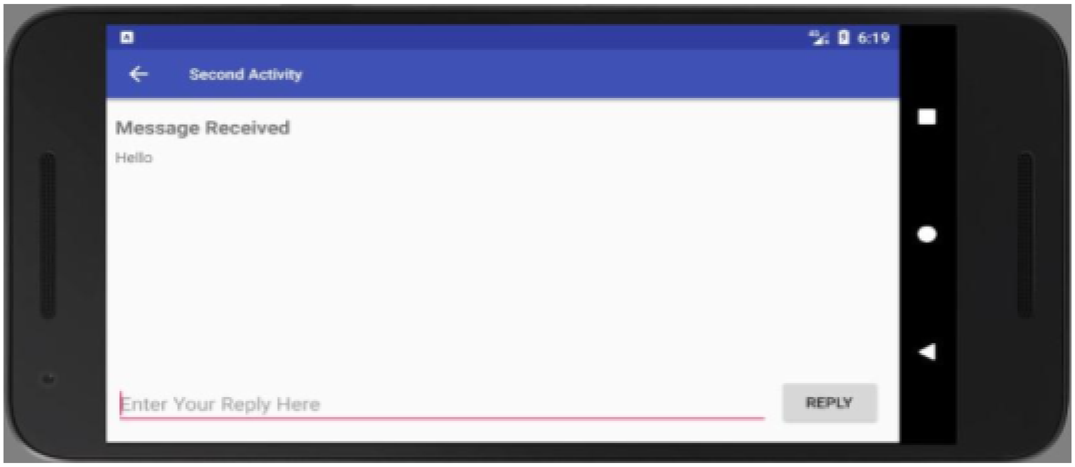
strings.xml

```
<resources>
    <string name="app_name">Two Activities</string>
    <string name="button_main">Send</string>
    <string name="editText_main">Enter Your Message Here</string>
    <string name="text_header">Message Received</string>
    <string name="button_second">Reply</string>
    <string name="editText_second">Enter Your Reply Here</string>
    <string name="text_header_reply">Reply Received</string>
</resources>
```

Output:



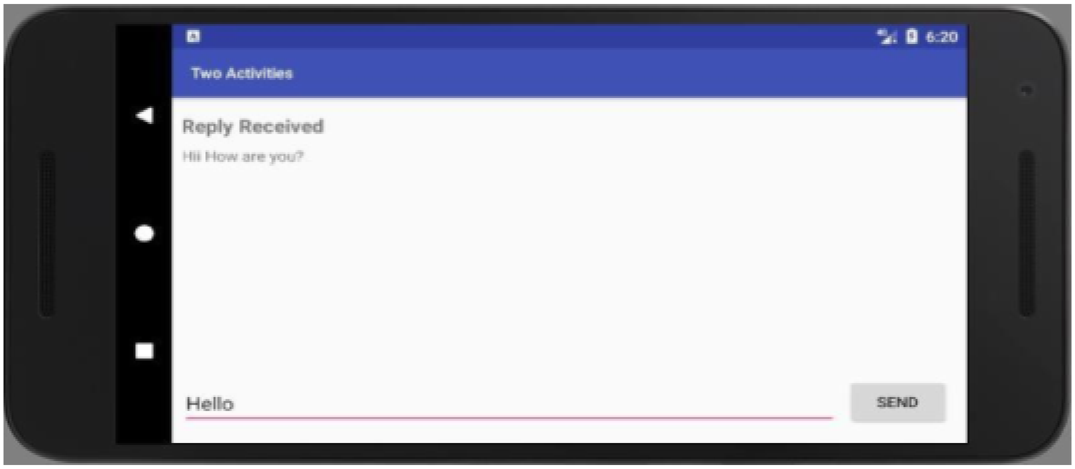
```
02-15 15:49:26.266 5878-5878/com.example.android.twoactivities D/MainActivity: -------
02-15 15:49:26.266 5878-5878/com.example.android.twoactivities D/MainActivity: onCreate
02-15 15:49:26.268 5878-5878/com.example.android.twoactivities D/MainActivity: onStart
02-15 15:49:26.271 5878-5878/com.example.android.twoactivities D/MainActivity: onResume
02-15 15:49:37.914 5878-5878/com.example.android.twoactivities D/MainActivity: Button clicked!
02-15 15:49:37.919 5878-5878/com.example.android.twoactivities D/MainActivity: onPause
02-15 15:49:37.953 5878-5878/com.example.android.twoactivities D/SecondActivity: onStart
02-15 15:49:37.955 5878-5878/com.example.android.twoactivities D/SecondActivity: onResume
02-15 15:49:38.430 5878-5878/com.example.android.twoactivities D/MainActivity: onStop
02-15 15:49:48.543 5878-5878/com.example.android.twoactivities D/SecondActivity: End SecondActivity
02-15 15:49:48.550 5878-5878/com.example.android.twoactivities D/SecondActivity: onPause
02-15 15:49:48.561 5878-5878/com.example.android.twoactivities D/MainActivity: onRestart
02-15 15:49:48.561 5878-5878/com.example.android.twoactivities D/MainActivity: onStart
02-15 15:49:48.562 5878-5878/com.example.android.twoactivities D/MainActivity: onResume
02-15 15:49:48.930 5878-5878/com.example.android.twoactivities D/SecondActivity: onStop
02-15 15:49:48.931 5878-5878/com.example.android.twoactivities D/SecondActivity: onDestroy
```



Message Received

Hello

Enter Your Reply Here



Reply Received

Hii How are you?

Hello

Practical: 5

Practical: Use an Options Menu

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
    tools:context="com.example.android.droidcafeoptions.MainActivity">

    <android.support.design.widget.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar            android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?attr/colorPrimary"
app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_main" />

    <android.support.design.widget.FloatingActionButton
android:id="@+id/fab"         android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="bottom|end"
android:layout_margin="16dp"
        app:srcCompat="@drawable/ic_shopping_cart" />

</android.support.design.widget.CoordinatorLayout>
```

content_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"     android:layout_height="match_parent"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="com.example.android.droidcafeoptions.MainActivity"
tools:showIn="@layout/activity_main">

   <TextView
        android:id="@+id/textintro"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"
android:text="@string/intro_text"
android:textSize="24sp"        android:textStyle="bold"
        app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

   <ImageView
        android:id="@+id/donut"        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:contentDescription="@string/donuts"
android:onClick="showDonutOrder"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textintro"
app:srcCompat="@drawable/donut_circle" />

   <ImageView
        android:id="@+id/ice_cream"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
        android:contentDescription="@string/ice_cream_sandwiches"
android:onClick="showIceCreamOrder"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/donut"
app:srcCompat="@drawable/icecream_circle" />

   <ImageView
        android:id="@+id/froyo"        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
```

```
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:contentDescription="@string/froyo"
android:onClick="showFroyoOrder"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ice_cream"
app:srcCompat="@drawable/froyo_circle" />

  <TextView
    android:id="@+id/donut_description"
android:layout_width="0dp"        android:layout_height="wrap_content"
android:layout_marginEnd="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"        android:text="@string/donuts"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/donut"
app:layout_constraintTop_toTopOf="@+id/donut" />

  <TextView
    android:id="@+id/ice_cream_description"
android:layout_width="0dp"        android:layout_height="wrap_content"
android:layout_marginEnd="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:text="@string/ice_cream_sandwiches"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/ice_cream"
app:layout_constraintTop_toTopOf="@+id/ice_cream" />

  <TextView
    android:id="@+id/froyo_description"
android:layout_width="0dp"        android:layout_height="wrap_content"
android:layout_marginEnd="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"        android:text="@string/froyo"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/froyo"
app:layout_constraintTop_toTopOf="@+id/froyo" />

</android.support.constraint.ConstraintLayout>
```

menu/menu_main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
   tools:context="com.example.android.droidcafeoptions.MainActivity">
   <item
      android:id="@+id/action_contact"        android:orderInCategory="100"
android:title="@string/action_contact"        app:showAsAction="never" />
   <item
      android:id="@+id/action_order"
android:icon="@drawable/ic_shopping_cart"
android:orderInCategory="10"        android:title="@string/action_order"
app:showAsAction="always" />
   <item
      android:id="@+id/action_status"
android:icon="@drawable/ic_status_info"        android:orderInCategory="20"
android:title="@string/action_status"        app:showAsAction="always" />
   <item
      android:id="@+id/action_favorites"
android:icon="@drawable/ic_favorite"        android:orderInCategory="30"
android:title="@string/action_favorites"        app:showAsAction="ifRoom" />
</menu>
```

MainActivity.java

```
package com.example.android.droidcafeoptions;
 import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.Toolbar; import android.view.Menu;
import android.view.MenuItem; import android.view.View;
import android.widget.Toast;

/**
* This app demonstrates images used as buttons and a floating action button to
* use an intent to launch a second activity. The app lets a user tap an image
* to make a choice. The app displays a Toast message showing the user's choice,
* and sends the choice as data with an intent to launch the second activity.
 *
* This version includes options in the options menu, in which some of the  * options
  appear as icons in the app bar.
 */
public class MainActivity extends AppCompatActivity {
  // Tag for the intent extra.
  public static final String EXTRA_MESSAGE =
         "com.example.android.droidcafeoptions.extra.MESSAGE";

  // The order message, displayed in the Toast and sent to the new Activity.
  private String mOrderMessage;

  /**
*  Creates the content view, the toolbar, and the floating action button.
*  This method is provided in the Basic Activity template.
   *
*  @param savedInstanceState Saved instance state bundle.
   */    @Override
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        Toolbar toolbar
= findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

       FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
             Intent intent = new Intent(MainActivity.this,
OrderActivity.class);
             intent.putExtra(EXTRA_MESSAGE, mOrderMessage);
startActivity(intent);
         }
       });
  }

  /**
*  Inflates the menu, and adds items to the action bar if it is present.
   *
*  @param menu Menu to inflate.
*  @return Returns true if the menu inflated.
   */    @Override
  public boolean onCreateOptionsMenu(Menu menu) {
       // Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.menu_main, menu);        return true;
  }

  /**
*  Handles app bar item clicks.
```

```
    *
* @param item Item clicked.
* @return True if one of the defined items was clicked.
    */    @Override
   public boolean onOptionsItemSelected(MenuItem item) {
switch (item.getItemId()) {            case R.id.action_order:
                Intent intent = new Intent(MainActivity.this,
OrderActivity.class);
                intent.putExtra(EXTRA_MESSAGE, mOrderMessage);
startActivity(intent);                return true;            case
R.id.action_status:
displayToast(getString(R.string.action_status_message));
return true;            case R.id.action_favorites:
displayToast(getString(R.string.action_favorites_message));
return true;            case R.id.action_contact:
displayToast(getString(R.string.action_contact_message));
return true;            default:                // Do nothing
        }
        return super.onOptionsItemSelected(item);
   }

   /**
* Displays a Toast with the message.
    *
* @param message Message to display.
    */
   public void displayToast(String message) {        Toast.makeText(getApplicationContext(),
message,
                Toast.LENGTH_SHORT).show();
   }

   /**
* Shows a message that the donut image was clicked.
    */
   public void showDonutOrder(View view) {
        mOrderMessage = getString(R.string.donut_order_message);
displayToast(mOrderMessage);
   }

   /**
* Shows a message that the ice cream sandwich image was clicked.
    */
   public void showIceCreamOrder(View view) {
        mOrderMessage = getString(R.string.ice_cream_order_message);
displayToast(mOrderMessage);
    }

    /**
* Shows a message that the froyo image was clicked.
     */
    public void showFroyoOrder(View view) {
        mOrderMessage = getString(R.string.froyo_order_message);
displayToast(mOrderMessage);
    }

}
```

activity_order.java

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
    tools:context="com.example.android.droidcafeoptions.OrderActivity">

    <TextView
        android:id="@+id/order_textview"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"        android:layout_marginTop="8dp"
android:text="@string/order_label_text"        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/name_label"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="32dp"
android:text="@string/name_label_text"
app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/order_textview" />

    <EditText
```

```
        android:id="@+id/name_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="8dp"
android:layout_marginStart="8dp"        android:ems="10"
        android:hint="@string/enter_name_hint"
android:inputType="textPersonName"
        app:layout_constraintBaseline_toBaselineOf="@+id/name_label"
app:layout_constraintStart_toEndOf="@+id/name_label" />

    <TextView
        android:id="@+id/address_label"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:text="@string/address_label_text"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/name_label" />

    <EditText
        android:id="@+id/address_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="8dp"
android:layout_marginStart="8dp"        android:ems="10"
        android:hint="@string/enter_address_hint"
android:inputType="textMultiLine"
        app:layout_constraintBaseline_toBaselineOf="@+id/address_label"
app:layout_constraintStart_toEndOf="@+id/address_label" />

    <TextView
        android:id="@+id/phone_label"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:text="@string/phone_label_string"
app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/address_text" />

    <EditText
    android:id="@+id/phone_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="8dp"
android:layout_marginStart="8dp"        android:ems="10"
    android:hint="@string/enter_phone_hint"        android:inputType="phone"
    app:layout_constraintBaseline_toBaselineOf="@+id/phone_label"
app:layout_constraintStart_toEndOf="@+id/phone_label" />

  <TextView
    android:id="@+id/note_label"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"
android:layout_marginTop="24dp"
android:text="@string/note_label_text"
app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/phone_label" />

  <EditText
    android:id="@+id/note_text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="8dp"        android:layout_marginStart="8dp"
android:ems="10"
    android:hint="@string/enter_note_hint"
android:inputType="textCapSentences|textMultiLine"
app:layout_constraintBaseline_toBaselineOf="@+id/note_label"
app:layout_constraintStart_toEndOf="@+id/note_label" />

  <TextView
    android:id="@+id/delivery_label"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"        android:layout_marginTop="24dp"
android:text="@string/choose_delivery_method"        android:textSize="18sp"
    app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/note_text" />

  <RadioGroup
    android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="24dp"
android:layout_marginStart="24dp"        android:orientation="vertical"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@id/delivery_label">

    <RadioButton          android:id="@+id/sameday"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
      android:text="@string/same_day_messenger_service" />

    <RadioButton          android:id="@+id/nextday"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
android:text="@string/next_day_ground_delivery"
android:checked="true"/>

    <RadioButton          android:id="@+id/pickup"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="onRadioButtonClicked"
android:text="@string/pick_up" />
  </RadioGroup>

</android.support.constraint.ConstraintLayout>
```

<u>OrderActivity.java</u>

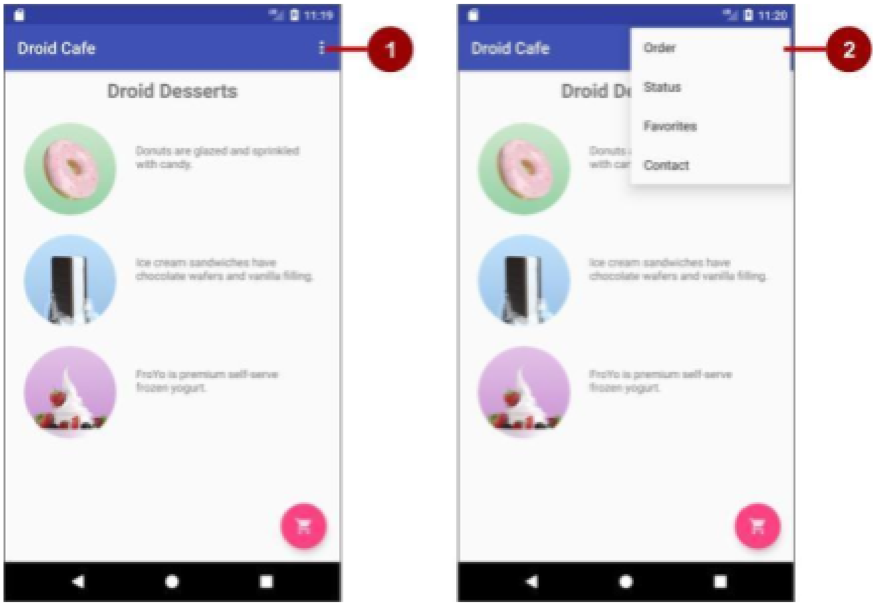
```
package com.example.android.droidcafeoptions;
 import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.RadioButton; import
android.widget.TextView; import
android.widget.Toast;

/**
* This activity handles radio buttons for choosing a delivery method for an  * order, and
  EditText input controls.
 */
public class OrderActivity extends AppCompatActivity {

  /**
* Sets the content view to activity_order, and gets the intent and its
* data.
   *
```

```
* @param savedInstanceState Saved instance state bundle.
    */     @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);         setContentView(R.layout.activity_order);

        // Get the intent and its data.
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
        TextView textView = findViewById(R.id.order_textview);
textView.setText(message);
    }

   /**
* Checks which radio button was clicked and displays a toast message to
* show the choice.
    *
* @param view The radio button view.
    */
   public void onRadioButtonClicked(View view) {
        // Is the button now checked?          boolean checked =
((RadioButton) view).isChecked();
        // Check which radio button was clicked.
switch (view.getId()) {              case R.id.sameday:
if (checked)
                    // Same day service
                    displayToast(getString(R.string.same_day_messenger_service));
                break;              case
R.id.nextday:                   if
(checked)
                    // Next day delivery
                    displayToast(getString(R.string.next_day_ground_delivery));                  break;
case R.id.pickup:                 if (checked)                     // Pick up
                    displayToast(getString(R.string.pick_up));
break;            default:                // Do nothing.                break;
        }
    }

   /**
* Displays the actual message in a toast message.
    *
* @param message Message to display.
    */
   public void displayToast(String message) {         Toast.makeText(getApplicationContext(),
message,
            Toast.LENGTH_SHORT).show();
    } }
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.android.droidcafeoptions">

    <application
        android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"        android:theme="@style/AppTheme">
        <activity
            android:name="com.example.android.droidcafeoptions.MainActivity"
android:label="@string/app_name"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".OrderActivity"
android:parentActivityName=".MainActivity"/>
    </application>
</manifest>
```

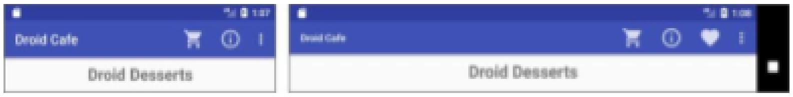
Output:

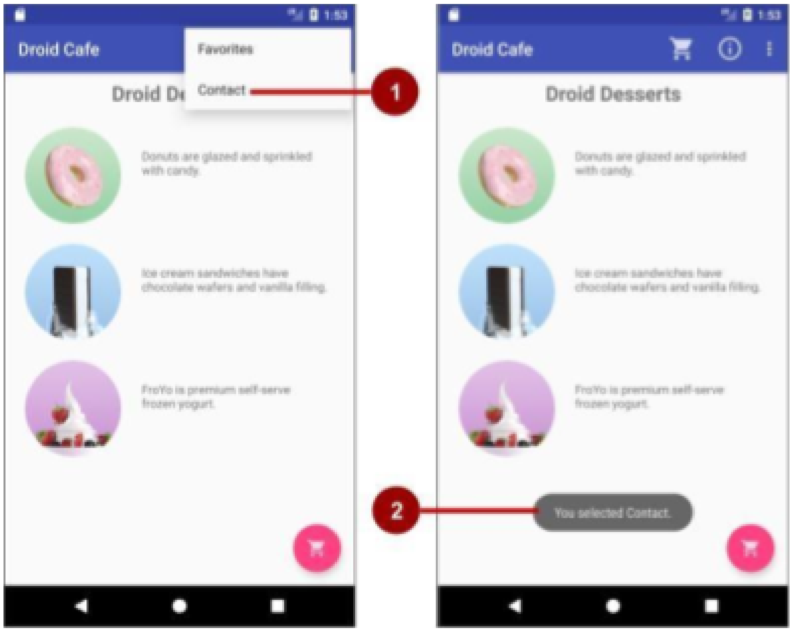


Donuts are glazed and sprinkled with candy.

Ice cream sandwiches have chocolate wafers and vanilla filling.

FroYo is premium self-serve frozen yogurt.





Donuts are glazed and sprinkled with candy.

Ice cream sandwiches have chocolate wafers and vanilla filling.

FroYo is premium self-serve frozen yogurt.

You selected Contact.

Practical: 6

Practical: Create a Recycler View and list the details of student using following fields:

- Create an activity that displays data in a RecyclerView
- Make the items in the list clickable
- Add a floating action button to add items to the list

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"    android:layout_height="match_parent"
   tools:context="com.example.android.recyclerview.MainActivity">

   <android.support.design.widget.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:theme="@style/AppTheme.AppBarOverlay">

      <android.support.v7.widget.Toolbar           android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?attr/colorPrimary"
app:popupTheme="@style/AppTheme.PopupOverlay" />

   </android.support.design.widget.AppBarLayout>

   <include layout="@layout/content_main" />

   <android.support.design.widget.FloatingActionButton
android:id="@+id/fab"        android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="bottom|end"
android:layout_margin="@dimen/fab_margin"
app:srcCompat="@drawable/ic_add_for_fab" />

</android.support.design.widget.CoordinatorLayout>
```

content_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
   app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="com.example.android.recyclerview.MainActivity"
tools:showIn="@layout/activity_main">

   <android.support.v7.widget.RecyclerView         android:id="@+id/recyclerview"
android:layout_width="match_parent"
android:layout_height="match_parent">
</android.support.v7.widget.RecyclerView>

</android.support.constraint.ConstraintLayout>
```

wordlist_item.xml

```
<?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="wrap_content"
android:orientation="vertical"     android:padding="6dp">

   <TextView
       android:id="@+id/word"        style="@style/word_title"
/>

</LinearLayout>
```

MainActivity.java

```
package com.example.android.recyclerview;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.LinearLayoutManager; import
android.support.v7.widget.RecyclerView; import
android.support.v7.widget.Toolbar; import android.view.Menu;
import android.view.MenuItem; import android.view.View;

import java.util.LinkedList;
```

```
/**
* Implements a basic RecyclerView that displays a list of generated words.
* - Clicking an item marks it as clicked.
* - Clicking the fab button adds a new word to the list.
 */
public class MainActivity extends AppCompatActivity {

   private final LinkedList<String> mWordList = new LinkedList<>();
    private RecyclerView mRecyclerView;
private WordListAdapter mAdapter;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        Toolbar toolbar
= findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

        FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {                int
wordListSize = mWordList.size();
                // Add a new word to the wordList.                mWordList.addLast("+ Word "
+ wordListSize);                // Notify the adapter, that the data has changed.
mRecyclerView.getAdapter().notifyItemInserted(wordListSize);
                // Scroll to the bottom.
mRecyclerView.smoothScrollToPosition(wordListSize);
            }
        });

        // Put initial data into the word list.        for (int i =
0; i < 20; i++) {            mWordList.addLast("Word " + i);
        }

        // Create recycler view.        mRecyclerView =
findViewById(R.id.recyclerview);        // Create an adapter and supply
the data to be displayed.
        mAdapter = new WordListAdapter(this, mWordList);
        // Connect the adapter with the recycler view.
        mRecyclerView.setAdapter(mAdapter);
        // Give the recycler view a default layout manager.
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

```
   }

   /**
* Inflates the menu, and adds items to the action bar if it is present.
    *
* @param menu Menu to inflate.
* @return Returns true if the menu inflated.
    */     @Override
   public boolean onCreateOptionsMenu(Menu menu) {
                  // Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.menu_main, menu);        return true;
   }

   /**
* Handles app bar item clicks.
    *
* @param item Item clicked.
* @return True if one of the defined items was clicked.
    */     @Override
   public boolean onOptionsItemSelected(MenuItem item) {         // Handle
action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        // This comment suppresses the Android Studio warning about simplifying
        // the return statements.
        //noinspection SimplifiableIfStatement         if (id
== R.id.action_settings) {            return true;
        }
        return super.onOptionsItemSelected(item);
   } }
```

WordListAdapter.java

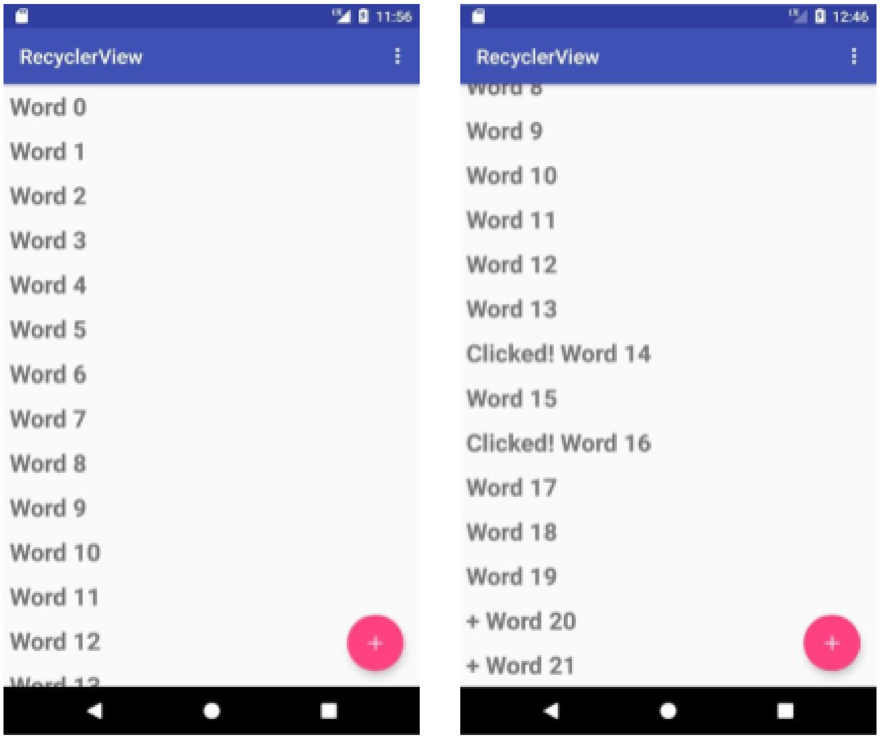
```
package com.example.android.recyclerview;

import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View; import
android.view.ViewGroup; import
android.widget.TextView;
 import java.util.LinkedList;
```

```
/**
*                     Shows how to implement a simple Adapter for a RecyclerView.
*                     Demonstrates how to add a click handler for each item in the
                      ViewHolder.
 */
public class WordListAdapter extends
     RecyclerView.Adapter<WordListAdapter.WordViewHolder> {

   private final LinkedList<String> mWordList;    private final
LayoutInflater mInflater;

   class WordViewHolder extends RecyclerView.ViewHolder
         implements View.OnClickListener {        public
final TextView wordItemView;        final
WordListAdapter mAdapter;

      /**
*                     Creates a new custom view holder to hold the view to display in
*                     the RecyclerView.
       *
*                     @param itemView The view in which to display the data.
*                     @param adapter The adapter that manages the the data and views
*                     for the RecyclerView.
       */
      public WordViewHolder(View itemView, WordListAdapter adapter) {
super(itemView);
         wordItemView = itemView.findViewById(R.id.word);
this.mAdapter = adapter;          itemView.setOnClickListener(this);
      }

      @Override
      public void onClick(View view) {
         // Get the position of the item that was clicked.          int
mPosition = getLayoutPosition();

         // Use that to access the affected item in mWordList.          String
element = mWordList.get(mPosition);          // Change the word in the
mWordList.

         mWordList.set(mPosition, "Clicked! " + element);          // Notify the
adapter, that the data has changed so it can
         // update the RecyclerView to display the data.
mAdapter.notifyDataSetChanged();
```

```
        }
    }
    public WordListAdapter(Context context, LinkedList<String> wordList) {
mInflater = LayoutInflater.from(context);          this.mWordList = wordList;
    }

    /**
*                              Called when RecyclerView needs a new ViewHolder of the given
                               type to
*                              represent an item.
     *
*                              This new ViewHolder should be constructed with a new View that
                               can
*                              represent the items of the given type. You can either create a new
                               View
*                              manually or inflate it from an XML layout file.
     *
*                              The new ViewHolder will be used to display items of the adapter
                               using
*                              onBindViewHolder(ViewHolder, int, List). Since it will be reused to
*                              display different items in the data set, it is a good idea to cache
*                              references to sub views of the View to avoid unnecessary
                               findViewById()
*                              calls.
     *
*                              @param parent   The ViewGroup into which the new View will be
                               added after
*                              it is bound to an adapter position.
*                              @param viewType The view type of the new View. @return A new
                               ViewHolder
*                              that holds a View of the given view type.
     */     @Override
        public WordListAdapter.WordViewHolder onCreateViewHolder(ViewGroup parent,
                                                                  int viewType) {
        // Inflate an item view.
        View mItemView = mInflater.inflate(
                R.layout.wordlist_item, parent, false);          return new
WordViewHolder(mItemView, this);
    }

    /**
*                              Called by RecyclerView to display the data at the specified position.
                               * This method should update the contents of the
                               ViewHolder.itemView to
```

```
*                    reflect the item at the given position.
   *
*                    @param holder   The ViewHolder which should be updated to
                     represent
*                    the contents of the item at the given position in the
*                    data set.
*                    @param position The position of the item within the adapter's data
                     set.
   */    @Override
  public void onBindViewHolder(WordListAdapter.WordViewHolder holder,
int position) {
      // Retrieve the data for that position.
      String mCurrent = mWordList.get(position);
      // Add the data to the view holder.        holder.wordItemView.setText(mCurrent);
  }

  /**
*                    Returns the total number of items in the data set held by the adapter.
   *
*                    @return The total number of items in this adapter.
   */    @Override
  public int getItemCount() {        return
mWordList.size();
  } }
```

Output:



11:56

RecyclerView

Word 0
Word 1
Word 2
Word 3
Word 4
Word 5
Word 6
Word 7
Word 8
Word 9
Word 10
Word 11
Word 12
Word 13

12:46

RecyclerView

Word 8
Word 9
Word 10
Word 11
Word 12
Word 13
Clicked! Word 14
Word 15
Clicked! Word 16
Word 17
Word 18
Word 19
+ Word 20
+ Word 21

Practical: 7

Practical: Theme, Custom Styles, Drawables

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
android:orientation="vertical"     android:padding="16dp"
   tools:context="com.example.android.scorekeeper.MainActivity">

   <RelativeLayout
       android:layout_width="match_parent"
android:layout_height="0dp"        android:layout_weight="1">

       <TextView
           android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:text="@string/team_1"            style="@style/TeamText" />

       <ImageButton
           android:id="@+id/decreaseTeam1"
android:layout_width="@dimen/button_size"
android:layout_height="@dimen/button_size"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_centerVertical="true"            style="@style/MinusButtons"
android:onClick="decreaseScore"/>

       <TextView
           android:id="@+id/score_1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:text="@string/initial_count"            style="@style/ScoreText" />

       <ImageButton
           android:id="@+id/increaseTeam1"
android:layout_width="@dimen/button_size"
android:layout_height="@dimen/button_size"
android:layout_alignParentEnd="true"
```

```
android:layout_alignParentRight="true"
android:layout_centerVertical="true"            style="@style/PlusButtons"
android:onClick="increaseScore"/>

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
android:layout_height="0dp"        android:layout_weight="1">

        <TextView
            android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:text="@string/team_2"            style="@style/TeamText" />

        <ImageButton
            android:id="@+id/decreaseTeam2"
android:layout_width="@dimen/button_size"
android:layout_height="@dimen/button_size"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_centerVertical="true"            style="@style/MinusButtons"
android:onClick="decreaseScore"/>

        <TextView            android:id="@+id/score_2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:text="@string/initial_count"
style="@style/ScoreText" />

        <ImageButton
            android:id="@+id/increaseTeam2"
android:layout_width="@dimen/button_size"
android:layout_height="@dimen/button_size"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_centerVertical="true"            style="@style/PlusButtons"
android:onClick="increaseScore"/>
    </RelativeLayout>

</LinearLayout>
```

MainActivity.java

```
package com.example.android.scorekeeper;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.AppCompatDelegate;
import android.view.Menu; import
android.view.MenuItem; import android.view.View;
import android.widget.TextView;

/***
 * Main Activity for the Scorekeeper app, which keeps score for any game  *
involving two teams. The user can increment or decrement the score  * for each
team using Button views.
 */
public class MainActivity extends AppCompatActivity {

   // Member variables for holding the score
private int mScore1;     private int mScore2;

   // Member variables for the two score TextView elements
private TextView mScoreText1;     private TextView mScoreText2;

   // Tags to be used as the keys in OnSavedInstanceState     static final
String STATE_SCORE_1 = "Team 1 Score";     static final String
STATE_SCORE_2 = "Team 2 Score";

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

       //Find the TextViews by ID        mScoreText1 =
findViewById(R.id.score_1);        mScoreText2 =
findViewById(R.id.score_2);

       // Restores the scores if there is savedInstanceState.        if
(savedInstanceState != null) {
                   mScore1 = savedInstanceState.getInt(STATE_SCORE_1);
              mScore2 = savedInstanceState.getInt(STATE_SCORE_2);

          //Set the score text views
          mScoreText1.setText(String.valueOf(mScore1));
mScoreText2.setText(String.valueOf(mScore2));
       }
   }

   /**
* Handles the onClick of both the decrement buttons.
    *
* @param view The button view that was clicked
    */
   public void decreaseScore(View view) {         // Get the
ID of the button that was clicked.         int viewID =
view.getId();        switch (viewID) {            // If it was on
Team 1:            case R.id.decreaseTeam1:
             // Decrement the score and update the TextView.                mScore1--;
             mScoreText1.setText(String.valueOf(mScore1));
break;           // If it was Team 2:            case R.id.decreaseTeam2:
             // Decrement the score and update the TextView.                mScore2--;
             mScoreText2.setText(String.valueOf(mScore2));
       }
   }

   /**
* Handles the onClick of both the increment buttons.
    *
* @param view The button view that was clicked
    */
   public void increaseScore(View view) {         // Get the ID
of the button that was clicked.         int viewID =
view.getId();        switch (viewID) {            // If it was on
Team 1:            case R.id.increaseTeam1:
             // Increment the score and update the TextView.                mScore1++;
             mScoreText1.setText(String.valueOf(mScore1));
break;           // If it was Team 2:            case R.id.increaseTeam2:
             // Increment the score and update the TextView.                mScore2++;
             mScoreText2.setText(String.valueOf(mScore2));
       }
   }

   /**
* Creates the night mode menu option.
    *
* @param menu The menu in the action bar
* @return True to display the menu, false to hide it
    */    @Override
   public boolean onCreateOptionsMenu(Menu menu) {
getMenuInflater().inflate(R.menu.main_menu, menu);        // Change the label of the
menu based on the state of the app.
       int nightMode = AppCompatDelegate.getDefaultNightMode();        if(nightMode
== AppCompatDelegate.MODE_NIGHT_YES){
           menu.findItem(R.id.night_mode).setTitle(R.string.day_mode);
       } else{
           menu.findItem(R.id.night_mode).setTitle(R.string.night_mode);
       }
       return true;
   }


   /**
* Handles options menu item clicks.
    *
* @param item The item that was pressed
* @return returns true since the item click wa handled
    */     @Override
   public boolean onOptionsItemSelected(MenuItem item) {
       // Check if the correct item was clicked.        if (item.getItemId() ==
R.id.night_mode) {           // Get the night mode state of the app.           int
nightMode = AppCompatDelegate.getDefaultNightMode();
           // Set the theme mode for the restarted activity.           if
(nightMode == AppCompatDelegate.MODE_NIGHT_YES) {
               AppCompatDelegate.setDefaultNightMode
                       (AppCompatDelegate.MODE_NIGHT_NO);
           } else {
               AppCompatDelegate.setDefaultNightMode
                       (AppCompatDelegate.MODE_NIGHT_YES);
           }
           // Recreate the activity for the theme change to take effect.           recreate();
       }
       return true;
   }

   /**
* Method that is called when the configuration changes,
* used to preserve the state of the app.
    *
* @param outState The bundle that will be passed in to the Activity when it is restored.
   */    @Override
   protected void onSaveInstanceState(Bundle outState) {
        // Save the scores.          outState.putInt(STATE_SCORE_1,
mScore1);          outState.putInt(STATE_SCORE_2, mScore2);
super.onSaveInstanceState(outState);
    } }
```

menu/main_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/night_mode"         android:title="@string/night_mode"/>
</menu>
```

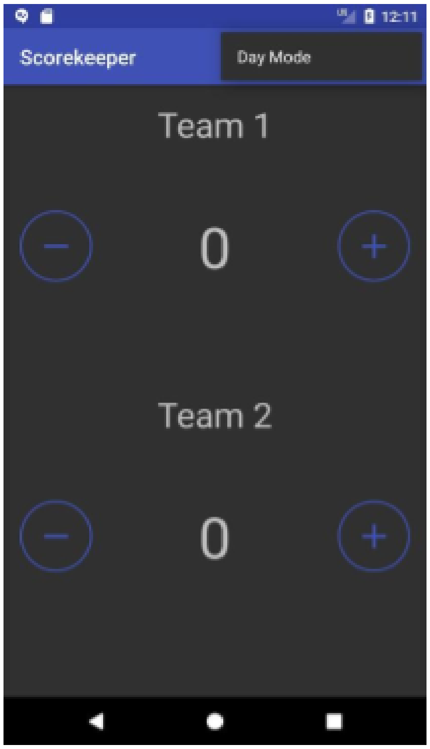
styles.xml

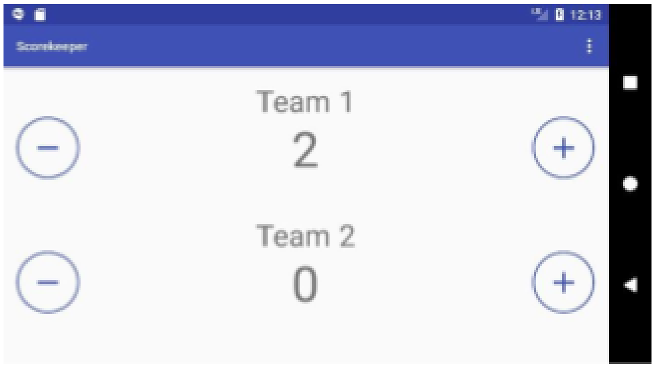
```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.DayNight.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

    <style name="ScoreButtons" parent="Widget.AppCompat.Button">
        <item name="android:background">@drawable/button_background</item>
        <item name="android:tint">@color/colorPrimary</item>
    </style>

    <style name="PlusButtons" parent="ScoreButtons">
        <item name="android:src">@drawable/ic_plus</item>
        <item name="android:contentDescription">
            @string/plus_button_description
        </item>
    </style>

    <style name="MinusButtons" parent="ScoreButtons">
        <item name="android:src">@drawable/ic_minus</item>
        <item name="android:contentDescription">
            @string/minus_button_description
        </item>
    </style>

    <style name="ScoreText">
    <item name="android:textAppearance">
      @style/TextAppearance.AppCompat.Display3
    </item>
  </style>

  <style name="TeamText">
    <item name="android:textAppearance">
      @style/TextAppearance.AppCompat.Display1
    </item>
  </style>

</resources>
```

strings.xml

```
<resources>
  <string name="app_name">Scorekeeper</string>
  <string name="team_1">Team 1</string>
  <string name="team_2">Team 2</string>
  <string name="initial_count">0</string>
  <string name="minus_button_description">Minus Button</string>
  <string name="plus_button_description">Plus Button</string>
  <string name="night_mode">Night Mode</string>
  <string name="day_mode">Day Mode</string>
</resources>
```

Output:


12:11
Scorekeeper
Day Mode
Team 1
0
Team 2
0


12:13
Scorekeeper
Team 1
2
Team 2
0

Practical: 8

Practical: Save user data in a database

activity_main.xml

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

   <TextView
       android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="SQLite
Database"        android:textSize="30sp"
       app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
    <Button
       android:id="@+id/add_contact"
android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="Add
Contact" />
    <Button
       android:id="@+id/update_contact"
android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="Update
Contact" />
    <Button
       android:id="@+id/delete_contact"
android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="Delete
Contact" />
    <Button
       android:id="@+id/get_contact"
android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="Get
Contact" />
    <Button
       android:id="@+id/all_contacts"        android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="Get all Contacts" />

</LinearLayout>
```

```
MainActivity.java package
com.example.sqlitedemoapp;

import androidx.appcompat.app.AppCompatActivity;
 import android.content.Intent;
import android.os.Bundle; import
android.util.Log; import
android.view.View; import
android.widget.Button;
 import java.io.FileNotFoundException;
import java.io.FileOutputStream; import
java.io.OutputStreamWriter; import
java.util.List;

public class MainActivity extends AppCompatActivity {

    Button add, update, delete, get, getall;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);         setContentView(R.layout.activity_main);
         add = findViewById(R.id.add_contact);         update
= findViewById(R.id.update_contact);         delete =
findViewById(R.id.delete_contact);         get =
findViewById(R.id.get_contact);         getall =
findViewById(R.id.all_contacts);

        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this, AddContactActivity.class);
startActivity(i);
            }         });
        update.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
UpdateContactActivity.class);                 startActivity(i);
            }         });
        delete.setOnClickListener(new View.OnClickListener() {            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
DeleteContactActivity.class);                 startActivity(i);
            }         });
        get.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this, GetContactActivity.class);
startActivity(i);
            }         });
        getall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(MainActivity.this,
GetAllContactActivity.class);                 startActivity(i);
            }
        });

    } }
```

Contact.java package

```
com.example.sqlitedemoapp;


    public class Contact {          int
_id;         String _name;
String _phone_number;
public Contact(){   }
        public Contact(int id, String name, String _phone_number){
this._id = id;              this._name = name;
            this._phone_number = _phone_number;
        }
        public Contact(String name, String _phone_number){              this._name
= name;
            this._phone_number = _phone_number;
        }
        public int getID(){            return
this._id;
        }
        public void setID(int id){            this._id =
id;
        }

        public String getName(){            return
this._name;
        }
        public void setName(String name){
this._name = name;
```

```
    }         public String
getPhoneNumber(){            return
this._phone_number;
    }
    public void setPhoneNumber(String phone_number){
this._phone_number = phone_number;
    }    }
```

DatabaseHandler.java

```
package
com.example.sqlitedemoapp;
 import android.content.ContentValues;
import android.content.Context; import
android.database.Cursor;
import android.database.sqlite.SQLiteDatabase; import
android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList; import
java.util.List;
  public class DatabaseHandler extends SQLiteOpenHelper {
private static final int DATABASE_VERSION = 1;
   private static final String DATABASE_NAME = "contactsManager";
private static final String TABLE_CONTACTS = "contacts";     private static
final String KEY_ID = "id";     private static final String KEY_NAME =
"name";     private static final String KEY_PH_NO = "phone_number";

   public DatabaseHandler(Context context) {
       super(context, DATABASE_NAME, null, DATABASE_VERSION);
       //3rd argument to be passed is CursorFactory instance
   }

   // Creating Tables     @Override
   public void onCreate(SQLiteDatabase db) {
       String CREATE_CONTACTS_TABLE = "CREATE TABLE " +
TABLE_CONTACTS + "("
              + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + KEY_NAME + "
TEXT,"
              + KEY_PH_NO + " TEXT" + ")";
db.execSQL(CREATE_CONTACTS_TABLE);
   }

   // Upgrading database
   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
       // Drop older table if existed
       db.execSQL("DROP TABLE IF EXISTS " + TABLE_CONTACTS);
```

```
          // Create tables again         onCreate(db);
     }

     // code to add the new contact      long
addContact(Contact contact) {
          SQLiteDatabase db = this.getWritableDatabase();

          ContentValues values = new ContentValues();         values.put(KEY_NAME,
contact.getName()); // Contact Name         values.put(KEY_PH_NO,
contact.getPhoneNumber()); // Contact Phone

          // Inserting Row
          long id =  db.insert(TABLE_CONTACTS, null, values);
//2nd argument is String containing nullColumnHack
db.close(); // Closing database connection         return id;
     }

     // code to get the single contact
     Contact getContact(int id) {
          SQLiteDatabase db = this.getReadableDatabase();

          Cursor cursor = db.query(TABLE_CONTACTS, new String[] { KEY_ID,
KEY_NAME, KEY_PH_NO }, KEY_ID + "=?",
                new String[] { String.valueOf(id) }, null, null, null, null);         if (cursor !=
null)            cursor.moveToFirst();

          Contact contact = new Contact(Integer.parseInt(cursor.getString(0)),
cursor.getString(1), cursor.getString(2));
          // return contact         return
contact;
     }

     // code to get all contacts in a list view      public
List<Contact> getAllContacts() {
          List<Contact> contactList = new ArrayList<Contact>();
          // Select All Query
          String selectQuery = "SELECT  * FROM " + TABLE_CONTACTS;

          SQLiteDatabase db = this.getWritableDatabase();
          Cursor cursor = db.rawQuery(selectQuery, null);
          // looping through all rows and adding to list         if
(cursor.moveToFirst()) {            do {
                Contact contact = new Contact();
```

```
            contact.setID(Integer.parseInt(cursor.getString(0)));
contact.setName(cursor.getString(1));               contact.setPhoneNumber(cursor.getString(2));
            // Adding contact to list
contactList.add(contact);           } while
(cursor.moveToNext());
      }

      // return contact list        return
contactList;
   }

   // code to update the single contact     public int
updateContact(Contact contact) {        SQLiteDatabase db =
this.getWritableDatabase();

      ContentValues values = new ContentValues();
values.put(KEY_NAME, contact.getName());        values.put(KEY_PH_NO,
contact.getPhoneNumber());

      // updating row
      return db.update(TABLE_CONTACTS, values, KEY_ID + " = ?",
new String[] { String.valueOf(contact.getID()) });
   }

   // Deleting single contact     public void
deleteContact(int id) {
      SQLiteDatabase db = this.getWritableDatabase();
db.delete(TABLE_CONTACTS, KEY_ID + " = ?",
new String[] { String.valueOf(id) });        db.close();
   }

   // Getting contacts Count     public int
getContactsCount() {
      String countQuery = "SELECT  * FROM " + TABLE_CONTACTS;
      SQLiteDatabase db = this.getReadableDatabase();        Cursor
cursor = db.rawQuery(countQuery, null);        cursor.close();

      // return count        return
cursor.getCount();
   }

}
```

activity_add_contact.xml

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

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".AddContactActivity">
    <EditText        android:id="@+id/name"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ems="10"        android:hint="Enter name"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.202"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.118" />
    <EditText        android:id="@+id/phno"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ems="10"        android:hint="Enter phone number"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.202"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.273" />
    <Button
       android:id="@+id/add_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="Add
Contact"        app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.141"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.427" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

AddContactActivity.java package
com.example.sqlitedemoapp;

```
import androidx.appcompat.app.AppCompatActivity;
```

```
 import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class AddContactActivity extends AppCompatActivity {

   EditText name, phno;
   Button b;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_add_contact);
        name = findViewById(R.id.name);
phno = findViewById(R.id.phno);         b =
findViewById(R.id.add_button);

       DatabaseHandler db = new DatabaseHandler(this);
       b.setOnClickListener(new View.OnClickListener() {
@Override
           public void onClick(View v) {                try
{
                   long id = db.addContact(new Contact(name.getText().toString(),
phno.getText().toString()));

                   Toast.makeText(getApplicationContext(), "Your contact has been saved successfully
with ID: " + id, Toast.LENGTH_SHORT).show();
               }catch(Exception e){
                   e.printStackTrace();
               }
           }
       });

   } }
```

activity_delete_contact.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".DeleteContactActivity">
```

```
   <EditText
       android:id="@+id/edit_id"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:ems="10"
       android:hint="Enter Contact ID" android:inputType="textPersonName"
             app:layout_constraintBottom_toBottomOf="parent"
       app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.297"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.113" />
    <Button
       android:id="@+id/delete_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="Delete
Contact"
       app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.226"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.225" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

<u>DeleteContactActivity.java</u> package com.example.sqlitedemoapp;

```
import androidx.appcompat.app.AppCompatActivity;
 import android.content.Intent;
import android.os.Bundle; import
android.view.View; import
android.widget.Button; import
android.widget.EditText; import
android.widget.Toast;

public class DeleteContactActivity extends AppCompatActivity {

   EditText id;
   Button b;

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

```
      id = findViewById(R.id.edit_id);         b =
findViewById(R.id.delete_button);

        DatabaseHandler db = new DatabaseHandler(this);

        b.setOnClickListener(new View.OnClickListener() {
@Override
            public void onClick(View v) {
                int cid = Integer.parseInt(id.getText().toString());

                try{
                    db.deleteContact(cid);
                    Toast.makeText(getApplicationContext(), "Contact has been deleted with ID: " + cid,
Toast.LENGTH_SHORT).show();
                }catch(Exception e){
                    e.printStackTrace();
                }
            }
        });
    } }
```

activity_update_contact.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".UpdateContactActivity">

    <EditText
        android:id="@+id/id_edit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:ems="10"
        android:hint="Enter Contact ID"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.181"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.083" />

    <EditText
```

```
        android:id="@+id/name_edit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:ems="10"        android:hint="Enter name"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.181"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.217" />

    <EditText
        android:id="@+id/phno_edit"        android:layout_width="wrap_content"
android:layout_height="wrap_content" android:ems="10"
        android:hint="Enter phone number"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.181"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.349" />
     <Button
        android:id="@+id/update_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="Update
Contact"        app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.111"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.484" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

UpdateContactActivity.java package com.example.sqlitedemoapp;

```
import androidx.appcompat.app.AppCompatActivity;
 import android.os.Bundle; import
android.view.View; import
android.widget.Button; import
android.widget.EditText; import
android.widget.Toast;

public class UpdateContactActivity extends AppCompatActivity {
   EditText id_edit, name_edit, phno_edit;
   Button update;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_update_contact);
       id_edit = findViewById(R.id.id_edit);
name_edit = findViewById(R.id.name_edit);
phno_edit = findViewById(R.id.phno_edit);        update
= findViewById(R.id.update_button);

      DatabaseHandler db = new DatabaseHandler(this);
       update.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            int cid = Integer.parseInt(id_edit.getText().toString());
            String cname = name_edit.getText().toString();              String
cphno = phno_edit.getText().toString();

            try{
                                    int id = db.updateContact(new Contact(cid, cname, cphno));
Toast.makeText(getApplicationContext(), "Contact has been updated with ID: " + id,
Toast.LENGTH_SHORT).show();

            }catch(Exception e){
               e.printStackTrace();
            }
         }
      });
   } }
```

activity_get_contact.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"    android:layout_height="match_parent"
tools:context=".GetContactActivity">

   <EditText
```

```
        android:id="@+id/edit_id_get"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:ems="10"
        android:hint="Enter Contact ID"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.257"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.074" />

    <Button
        android:id="@+id/button_get"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="Get
Contact"
        app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.179"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.203" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

GetContactActivity.java

```
package com.example.sqlitedemoapp;

import androidx.appcompat.app.AppCompatActivity;
 import android.os.Bundle; import
android.view.View; import
android.widget.Button; import
android.widget.EditText; import
android.widget.Toast;

public class GetContactActivity extends AppCompatActivity {

    EditText id;
    Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_get_contact);
        id = findViewById(R.id.edit_id_get);         b
= findViewById(R.id.button_get);

       DatabaseHandler db = new DatabaseHandler(this);

       b.setOnClickListener(new View.OnClickListener() {
@Override
          public void onClick(View v) {
             int cid =  Integer.parseInt(id.getText().toString());               try {
                Contact c = db.getContact(cid);
                Toast.makeText(getApplicationContext(), "Name: " + c.getName() +
"\n" + "Phone number: " + c.getPhoneNumber(), Toast.LENGTH_SHORT).show();
             }catch(Exception e){
                Toast.makeText(getApplicationContext(), "Not found",
Toast.LENGTH_SHORT).show();
             }
          }
       });
    } }
```

activity_get_all_contact.xml

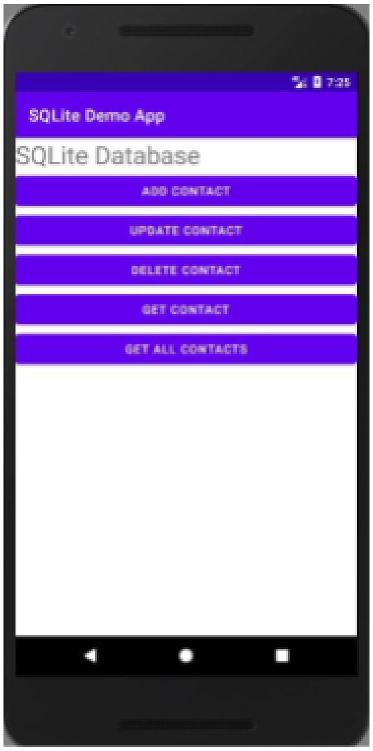
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".GetAllContactActivity">
    <Button
       android:id="@+id/button_getall"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:text="Get All
Contacts"        app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.164"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.087" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

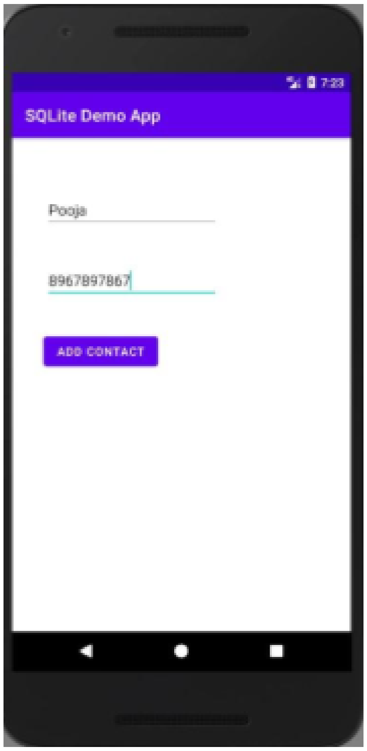
GetAllContactActivity.java package com.example.sqlitedemoapp;

```
import androidx.appcompat.app.AppCompatActivity;
```

```
 import android.os.Bundle; import
android.util.Log; import
android.view.View; import
android.widget.Button; import
android.widget.TextView; import
android.widget.Toast;
 import java.util.ArrayList;
import java.util.List;

public class GetAllContactActivity extends AppCompatActivity {

    Button b;

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

b  = findViewById(R.id.button_getall);

        DatabaseHandler db = new DatabaseHandler(this);

        b.setOnClickListener(new View.OnClickListener() {
@Override
            public void onClick(View v) {
String c = "";                try{
                    List<Contact> contacts = db.getAllContacts();

                    for (Contact cn : contacts) {
c  = "Id: " + cn.getID() + ", Name: " + cn.getName() + ", Phone: " +
                            cn.getPhoneNumber();
                        Toast.makeText(getApplicationContext(), c,
Toast.LENGTH_SHORT).show();
                    }
                }catch(Exception e){
                    e.printStackTrace();
                }
            }
        });
    } }
```

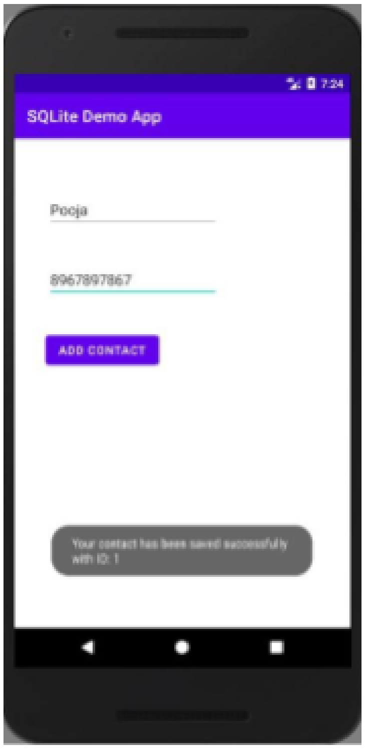
Output:





Pooja

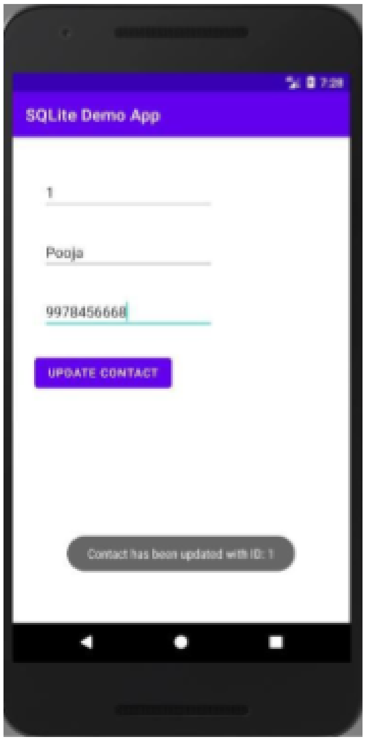
8967897867



Pooja

8967897867

Your contact has been saved successfully with ID: 1

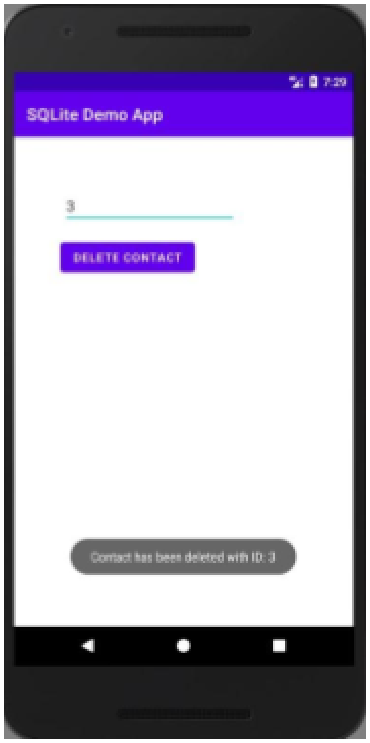


1

Pooja

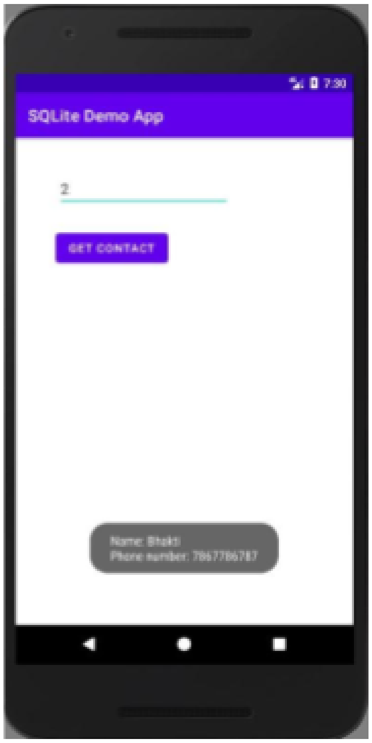
9978456668

Contact has been updated with ID: 1



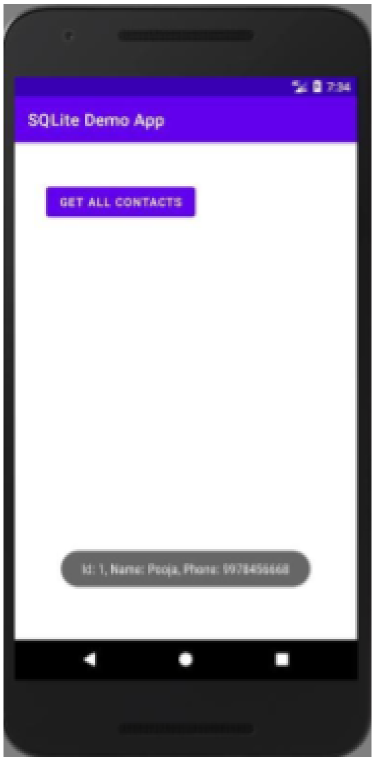
3

Contact has been deleted with ID: 3

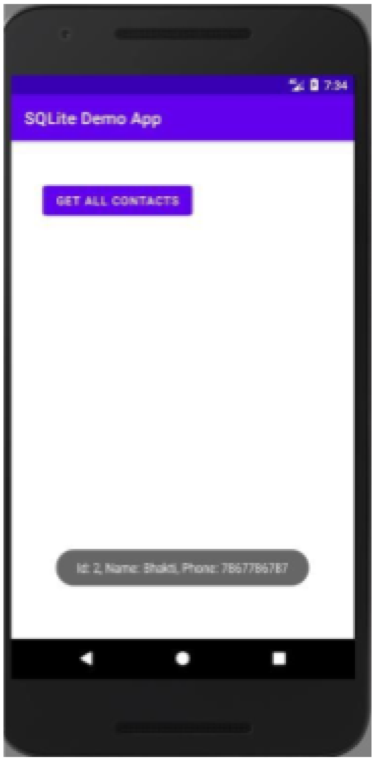


2

Name: Bhakti
Phone number: 7867786787



```
Id: 1, Name: Pooja, Phone: 9978456668
```



```
Id: 2, Name: Bhakti, Phone: 7867786787
```

Practical: 9

Practical: Use an AsyncTask to access remote database

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="16dp"     tools:context=".MainActivity">

   <TextView
       android:id="@+id/instructions"        android:layout_width="match_parent"
android:layout_height="wrap_content"        android:text="@string/instructions"
       android:textAppearance="@style/TextAppearance.AppCompat.Title"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"/>

   <EditText
       android:id="@+id/bookInput"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"        android:hint="@string/input_hint"
android:inputType="text"
       app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/instructions"/>
    <Button
       android:id="@+id/searchButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="8dp"        android:onClick="searchBooks"
android:text="@string/button_text"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/bookInput"/>

   <TextView
       android:id="@+id/titleText"        android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:layout_marginTop="16dp"
android:textAppearance="@style/TextAppearance.AppCompat.Headline"
app:layout_constraintStart_toStartOf="parent"
       app:layout_constraintTop_toBottomOf="@+id/searchButton"/>
   <TextView
       android:id="@+id/authorText"         android:layout_width="wrap_content"
android:layout_height="wrap_content"         android:layout_marginTop="8dp"
       android:textAppearance="@style/TextAppearance.AppCompat.Headline"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/titleText"/>

</android.support.constraint.ConstraintLayout>
```

NetworkUtils.java

```
package com.example.android.whowroteit;
 import android.net.Uri;
import android.util.Log;
 import java.io.BufferedReader;
import java.io.IOException; import
java.io.InputStream; import
java.io.InputStreamReader; import
java.net.HttpURLConnection; import
java.net.URL;

/**
 * Utility class for using the Google Book Search API to download book  *
information.
 */
public class NetworkUtils {

   private static final String LOG_TAG = NetworkUtils.class.getSimpleName();
   // Constants for the various components of the Books API request.
   //
   // Base endpoint URL for the Books API.     private
static final String BOOK_BASE_URL =
          "https://www.googleapis.com/books/v1/volumes?";
   // Parameter for the search string.     private static
final String QUERY_PARAM = "q";
   // Parameter that limits search results.
   private static final String MAX_RESULTS = "maxResults";
   // Parameter to filter by print type.
   private static final String PRINT_TYPE = "printType";


   /**
*  Static method to make the actual query to the Books API.
    *
```

```
* @param queryString the query string.
* @return the JSON response string from the query.
   */
  static String getBookInfo(String queryString) {

    // Set up variables for the try block that need to be closed in the
    // finally block.
    HttpURLConnection urlConnection = null;
    BufferedReader reader = null;
    String bookJSONString = null;

    try {
      // Build the full query URI, limiting results to 10 items and
      // printed books.
      Uri builtURI = Uri.parse(BOOK_BASE_URL).buildUpon()
          .appendQueryParameter(QUERY_PARAM, queryString)
          .appendQueryParameter(MAX_RESULTS, "10")
          .appendQueryParameter(PRINT_TYPE, "books")
          .build();

      // Convert the URI to a URL,
      URL requestURL = new URL(builtURI.toString());

      // Open the network connection.         urlConnection =
(HttpURLConnection) requestURL.openConnection();
urlConnection.setRequestMethod("GET");         urlConnection.connect();

      // Get the InputStream.
      InputStream inputStream = urlConnection.getInputStream();

      // Create a buffered reader from that input stream.         reader = new
BufferedReader(new InputStreamReader(inputStream));
      // Use a StringBuilder to hold the incoming response.
      StringBuilder builder = new StringBuilder();

      String line;
      while ((line = reader.readLine()) != null) {
        // Add the current line to the string.          builder.append(line);

        // Since this is JSON, adding a newline isn't necessary (it won't          // affect
parsing) but it does make debugging a *lot* easier
        // if you print out the completed buffer for debugging.
builder.append("\n");
      }
```

```
             if (builder.length() == 0) {
               // Stream was empty.  Exit without parsing.                return null;
            }

            bookJSONString = builder.toString();

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            // Close the connection and the buffered reader.              if
(urlConnection != null) {                 urlConnection.disconnect();
            }
            if (reader != null) {                 try {
                    reader.close();                 } catch
(IOException e) {
                    e.printStackTrace();
                }
            }
        }

        // Write the final JSON response to the log
        Log.d(LOG_TAG, bookJSONString);

        return bookJSONString;
    } }
```

FetchBook.java

```
package com.example.android.whowroteit;
 import android.os.AsyncTask;
import android.widget.TextView;
 import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
 import java.lang.ref.WeakReference;

/**
 * FetchBook is an AsyncTask implementation that opens a network connection  * and
queryies the Book Service API.
 */
public class FetchBook extends AsyncTask<String, Void, String> {

    // Variables for the results TextViews.
    // These are WeakReferences to prevent "leaky context" -- weak references
```

```
   // enable the activity to be garbage collected if it is not needed.
   private WeakReference<TextView> mTitleText;     private
WeakReference<TextView> mAuthorText;

   // Constructor, provides references to the views in MainActivity.
   FetchBook(TextView titleText, TextView authorText) {        this.mTitleText =
new WeakReference<>(titleText);         this.mAuthorText = new
WeakReference<>(authorText);
   }

   /**
*              Use the getBookInfo() method in the NetworkUtils class to make
*              the connection in the background.
    *
*              @param strings String array containing the search data.      * @return
               Returns the JSON string from the Books API, or
*              null if the connection failed.
    */    @Override
   protected String doInBackground(String... strings) {        return
NetworkUtils.getBookInfo(strings[0]);
   }

   /**
*              Handles the results on the UI thread. Gets the information from
*              the JSON result and updates the views.
    *
*              @param s Result from the doInBackground() method containing the raw
*              JSON response, or null if it failed.
    */    @Override
   protected void onPostExecute(String s) {        super.onPostExecute(s);
       try {
          // Convert the response into a JSON object.            JSONObject
jsonObject = new JSONObject(s);
          // Get the JSONArray of book items.
          JSONArray itemsArray = jsonObject.getJSONArray("items");

          // Initialize iterator and results fields.
          int i = 0;           String title =
null;
          String authors = null;

          // Look for results in the items array, exiting when both the            // title and
author are found or when all items have been checked.
          while (i < itemsArray.length() &&
          (authors == null && title == null)) {
        // Get the current item information.
        JSONObject book = itemsArray.getJSONObject(i);
        JSONObject volumeInfo = book.getJSONObject("volumeInfo");

        // Try to get the author and title from the current item,
        // catch if either field is empty and move on.                try {
          title = volumeInfo.getString("title");                    authors =
volumeInfo.getString("authors");                } catch (JSONException e) {
          e.printStackTrace();
        }

        // Move to the next item.                i++;
      }

      // If both are found, display the result.            if (title !=
null && authors != null) {
mTitleText.get().setText(title);
mAuthorText.get().setText(authors);
      } else {
        // If none are found, update the UI to show failed results.
        mTitleText.get().setText(R.string.no_results);
mAuthorText.get().setText("");
      }

    } catch (Exception e) {
      // If onPostExecute() does not receive a proper JSON string,
      // update the UI to show failed results.
      mTitleText.get().setText(R.string.no_results);
      mAuthorText.get().setText("");
      e.printStackTrace();
    }

  } }
```

ActivityMain.java

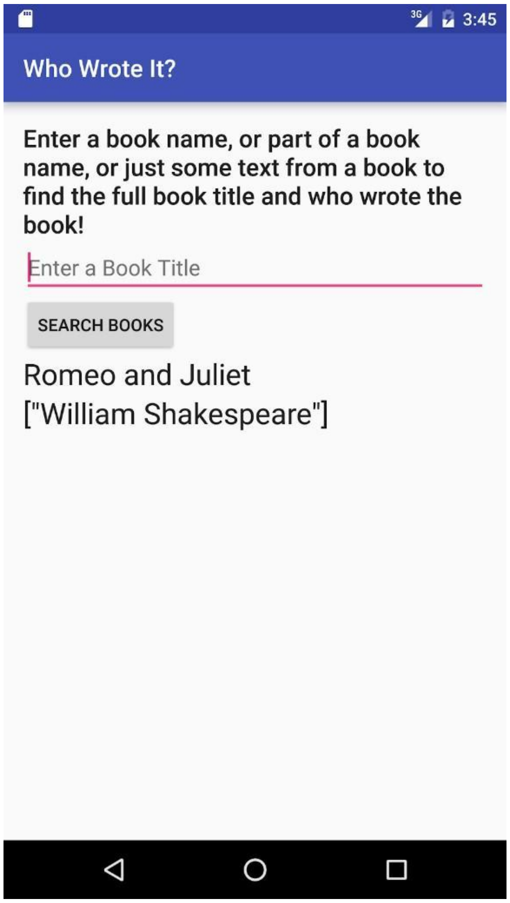
```
package com.example.android.whowroteit;
 import android.content.Context; import
android.net.ConnectivityManager; import
android.net.NetworkInfo;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText; import
android.widget.TextView;
```

```
/**
* The WhoWroteIt app queries the Book Search API for books based
* on a user's search. It uses an AsyncTask to run the search task in  * the
  background.
 */ public class MainActivity extends AppCompatActivity {

  // Variables for the search input field and results TextViews.
  private EditText mBookInput;
  private TextView mTitleText;    private
TextView mAuthorText;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);      setContentView(R.layout.activity_main);
     mBookInput = findViewById(R.id.bookInput);
mTitleText = findViewById(R.id.titleText);
mAuthorText = findViewById(R.id.authorText);
  }

  /**
* onClick handler for the "Search Books" button.
   *
* @param view The view (Button) that was clicked.
   */
  public void searchBooks(View view) {
    // Get the search string from the input field.
    String queryString = mBookInput.getText().toString();

    // Hide the keyboard when the button is pushed.
InputMethodManager inputManager = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);      if
(inputManager != null) {
      inputManager.hideSoftInputFromWindow(view.getWindowToken(),
          InputMethodManager.HIDE_NOT_ALWAYS);
    }

    // Check the status of the network connection.      ConnectivityManager
connMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = null;      if
(connMgr != null) {
      networkInfo = connMgr.getActiveNetworkInfo();
    }
    // If the network is available, connected, and the search field
    // is not empty, start a FetchBook AsyncTask.
    if (networkInfo != null && networkInfo.isConnected()
        && queryString.length() != 0) {
      new FetchBook(mTitleText, mAuthorText).execute(queryString);
mAuthorText.setText("");
      mTitleText.setText(R.string.loading);
    }
    // Otherwise update the TextView to tell the user there is no
    // connection, or no search term.        else {
      if (queryString.length() == 0) {            mAuthorText.setText("");
        mTitleText.setText(R.string.no_search_term);           } else
{
        mAuthorText.setText("");
        mTitleText.setText(R.string.no_network);
      }
    }
  }
}
```

strings.xml

```
<resources>
  <string name="app_name">WhoWroteIt</string>

  <!-- Strings in activity_main view -->
  <string name="instructions">Enter a book name, or  part of the book name, or just text from a
book to find the full book title and who wrote the book! </string>
  <string name="button_text">Search Books</string>
  <string name="input_hint">Book Title</string>

  <!-- User messages for in-process queries and error results. -->
  <string name="loading">Loading…</string>
  <string name="no_results">"No Results Found"</string>
  <string name="no_search_term">Please enter a search term</string>     <string
name="no_network">Please check your network connection and try again.</string>
</resources>
```

Output:



Who Wrote It?

Enter a book name, or part of a book name, or just some text from a book to find the full book title and who wrote the book!

Enter a Book Title

SEARCH BOOKS

Romeo and Juliet
["William Shakespeare"]

Practical: 10

Practical: Use Firebase to perform CRUID operation

User.java package

```
info.androidhive.firebase;

import com.google.firebase.database.IgnoreExtraProperties;

/**
* Created by Ravi Tamada on 07/10/16.
* www.androidhive.info
 */

@IgnoreExtraProperties
public class User {

   public String name;
public String email;

   // Default constructor required for calls
to     // DataSnapshot.getValue(User.class)
   public User() {
   }

   public User(String name, String email) {
this.name = name;
      this.email = email;
   }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
   android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="info.androidhive.firebase.MainActivity">

    <TextView
        android:id="@+id/txt_user"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:paddingBottom="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_horizontal_margin"
        android:textSize="20dp" />

    <LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical">

        <android.support.design.widget.TextInputLayout
android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/name"
android:inputType="textCapWords"
android:maxLines="1" />

        </android.support.design.widget.TextInputLayout>

        <android.support.design.widget.TextInputLayout
android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
android:id="@+id/email"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/email"
android:inputType="textEmailAddress"
android:maxLines="1" />

        </android.support.design.widget.TextInputLayout>
        <Button
android:id="@+id/btn_save"
style="?android:textAppearanceSmall"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:background="@color/colorPrimary"
android:text="@string/action_save"
android:textColor="@android:color/white"
android:textStyle="bold" />

    </LinearLayout>

</LinearLayout>
```

MainActivity.java package

```
info.androidhive.firebase;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import
android.text.TextUtils;
import android.util.Log;
import android.view.View;
import
android.widget.Button;
import
android.widget.EditText;
import android.widget.TextView;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import
com.google.firebase.database.DatabaseReference;
import
com.google.firebase.database.FirebaseDatabase;
import
com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {
    private static final String TAG =
MainActivity.class.getSimpleName();     private TextView
txtDetails;     private EditText inputName, inputEmail;     private
Button btnSave;     private DatabaseReference mFirebaseDatabase;
    private FirebaseDatabase mFirebaseInstance;

    private String userId;

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

        // Displaying toolbar icon
        getSupportActionBar().setDisplayShowHomeEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        txtDetails = (TextView) findViewById(R.id.txt_user);
inputName = (EditText) findViewById(R.id.name);
inputEmail = (EditText) findViewById(R.id.email);
btnSave = (Button) findViewById(R.id.btn_save);
mFirebaseInstance = FirebaseDatabase.getInstance();

        // get reference to 'users' node
        mFirebaseDatabase = mFirebaseInstance.getReference("users");

        // store app title to 'app_title' node
        mFirebaseInstance.getReference("app_title").setValue("Realtime Database");

        // app_title change listener
        mFirebaseInstance.getReference("app_title").addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
Log.e(TAG, "App title updated");

                String appTitle = dataSnapshot.getValue(String.class);

                // update toolbar title
                getSupportActionBar().setTitle(appTitle);
            }

            @Override
            public void onCancelled(DatabaseError error) {
```

```
        // Failed to read value
        Log.e(TAG, "Failed to read app title value.", error.toException());
      }
    });

    // Save / update the user
    btnSave.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        String name = inputName.getText().toString();
        String email = inputEmail.getText().toString();

        // Check for already existed
userId          if
(TextUtils.isEmpty(userId)) {
createUser(name, email);
        } else {
          updateUser(name, email);
        }
      }
    });

    toggleButton();
  }

  // Changing button text
private void toggleButton() {
if (TextUtils.isEmpty(userId))
{
btnSave.setText("Save");
    } else {
      btnSave.setText("Update");
    }
  }

  /**
   * Creating new user node under 'users'
   */
  private void createUser(String name, String email) {
    // TODO
    // In real apps this userId should be
fetched     // by implementing firebase
auth     if (TextUtils.isEmpty(userId)) {
      userId = mFirebaseDatabase.push().getKey();
```

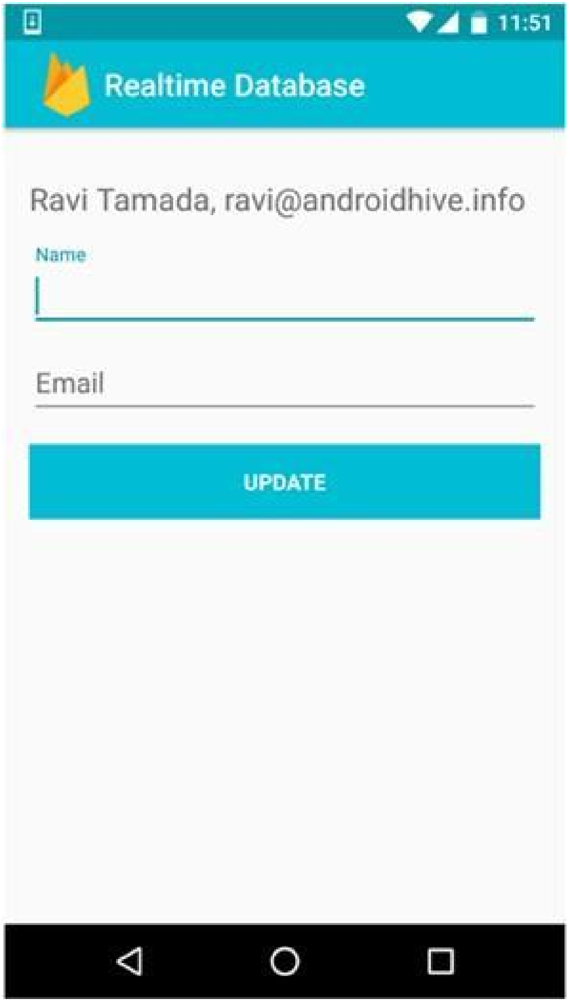
```
        }

        User user = new User(name, email);

        mFirebaseDatabase.child(userId).setValue(user);

        addUserChangeListener();
    }

    /**
     * User data change listener
     */
    private void addUserChangeListener() {
        // User data change listener
        mFirebaseDatabase.child(userId).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                User user = dataSnapshot.getValue(User.class);

                // Check for
null                    if (user ==
null) {
                    Log.e(TAG, "User data is null!");
return;
                }

                Log.e(TAG, "User data is changed!" + user.name + ", " + user.email);

                // Display newly updated name and email
txtDetails.setText(user.name + ", " + user.email);

                // clear edit text
inputEmail.setText("");
inputName.setText("");
toggleButton();
            }

            @Override
            public void onCancelled(DatabaseError error) {
                // Failed to read value
                Log.e(TAG, "Failed to read user", error.toException());
            }
        });
```

```
   }

   private void updateUser(String name, String email) {
       // updating the user via child
nodes         if
(!TextUtils.isEmpty(name))
           mFirebaseDatabase.child(userId).child("name").setValue(name);

       if (!TextUtils.isEmpty(email))
           mFirebaseDatabase.child(userId).child("email").setValue(email);
   }
}
```

JSON Structure:

```
{
  "app_title" : "Realtime Database",
  "users" : {
    "-KTYWvZG4Qn9ZYTc47O6" : {
      "email" : "ravi@androidhive.info",
      "name" : "Ravi Tamada"
    }
  } }
```

Output:



Ravi Tamada, ravi@androidhive.info

Practical: 11

Practical: BroadcastReceiver

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".MainActivity">

   <Button
       android:id = "@+id/sendBroadcast"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/send_custom_broadcast"
android:onClick="sendCustomBroadcast"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
package
com.android.fundamentals.powerreceiver;
 import android.content.Intent; import
android.content.IntentFilter;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.app.AppCompatActivity; import
android.os.Bundle; import android.view.View;

/**
* The Power Receiver app responds to system broadcasts about the power
* connected state as well as a custom broadcast that is sent when the user  * taps the
   button.
 */
public class MainActivity extends AppCompatActivity {
    private CustomReceiver mReceiver = new CustomReceiver();

   // String constant that defines the custom broadcast Action.     private static
final String ACTION_CUSTOM_BROADCAST =
          BuildConfig.APPLICATION_ID + ".ACTION_CUSTOM_BROADCAST";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);         setContentView(R.layout.activity_main);

        // Define the IntentFilter.
        IntentFilter filter = new IntentFilter();
        // Add system broadcast actions sent by the system when the power is
        // connected and disconnected.
filter.addAction(Intent.ACTION_POWER_CONNECTED);
filter.addAction(Intent.ACTION_POWER_DISCONNECTED);

        // Register the receiver using the activity context, passing in the
        // IntentFilter.         this.registerReceiver(mReceiver, filter);

        // Register the receiver to receive custom broadcast.
        LocalBroadcastManager.getInstance(this).registerReceiver
                (mReceiver, new IntentFilter(ACTION_CUSTOM_BROADCAST));
    }

    /**
*  Click event handler for the button, that sends custom broadcast using the
*  LocalBroadcastManager.
     */
    public void sendCustomBroadcast(View view) {
        Intent customBroadcastIntent = new Intent(ACTION_CUSTOM_BROADCAST);
        LocalBroadcastManager.getInstance(this)
                .sendBroadcast(customBroadcastIntent);
    }

    /**
*  Unregisters the broadcast receivers when the app is destroyed.
     */     @Override
    protected void onDestroy() {          //
Unregister the receivers.
        this.unregisterReceiver(mReceiver);
        LocalBroadcastManager.getInstance(this).unregisterReceiver(mReceiver);
super.onDestroy();
    } }
```

CustomReceiver.java

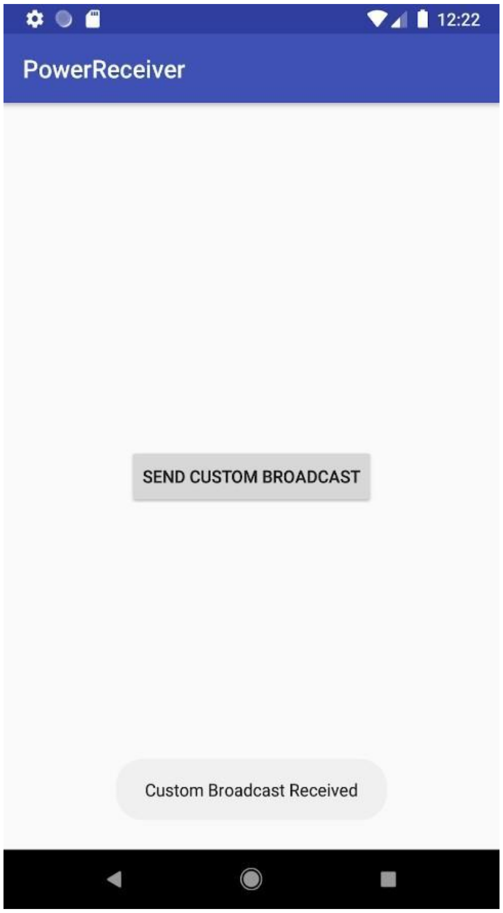
```
/*
*                              Copyright (C) 2018 Google Inc.
 *
```

```
*                   Licensed under the Apache License, Version 2.0 (the "License");  *
                    you may not use this file except in compliance with the License.
*                   You may obtain a copy of the License at
 *
*                   http://www.apache.org/licenses/LICENSE-2.0
 *
*                   Unless required by applicable law or agreed to in writing, software
*                   distributed under the License is distributed on an "AS IS" BASIS,
*                   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
                    express or implied.  * See the License for the specific language
                    governing permissions and  * limitations under the License.
 */
package com.android.fundamentals.powerreceiver;
 import android.arch.core.BuildConfig; import
android.content.BroadcastReceiver; import
android.content.Context; import
android.content.Intent; import
android.widget.Toast;

/**
*                   Broadcast Receiver implementation that delivers a custom Toast  *
                    message when it receives any of the registered broadcasts.
 */
public class CustomReceiver extends BroadcastReceiver {

   // String constant that defines the custom broadcast Action.    private static
final String ACTION_CUSTOM_BROADCAST =
         BuildConfig.APPLICATION_ID +
".ACTION_CUSTOM_BROADCAST";

   /**
*                   This callback method gets called when the Broadcast Receiver
                    receives a
*                   broadcast that it is registered for.
    *
*                   @param context The context in which broadcast receiver is running.
*                   @param intent The broadcast is delivered in the form of an intent
                    which
*                   contains the broadcast action.
    */    @Override
   public void onReceive(Context context, Intent intent) {
      String intentAction = intent.getAction();

      if (intentAction != null) {
          String toastMessage = context.getString(R.string.unknown_action);
          switch (intentAction){                case
Intent.ACTION_POWER_CONNECTED:
                toastMessage = context.getString(R.string.power_connected);
break;                case Intent.ACTION_POWER_DISCONNECTED:
                toastMessage =
                      context.getString(R.string.power_disconnected);                 break;
case ACTION_CUSTOM_BROADCAST:                     toastMessage =
                      context.getString(R.string.custom_broadcast_toast);                 break;
          }

          Toast.makeText(context, toastMessage, Toast.LENGTH_SHORT).show();
       }
   } }
```

strings.xml

```
<resources>
   <string name="app_name">PowerReceiver</string>
   <string name="power_connected">Power connected!</string>
   <string name="power_disconnected">Power disconnected!</string>
   <string name="custom_broadcast_toast">Custom Broadcast Received</string>
   <string name="send_custom_broadcast">Send Custom Broadcast</string>
   <string name="receiver_on">Receiver On</string>
   <string name="receiver_off">Receiver Off</string>
   <string name="unknown_action">unknown intent action</string>
</resources>
```

Output:



Custom Broadcast Received

Practical: 12

Practical: Notifications

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"    android:layout_height="match_parent"
tools:context=".MainActivity">
   <Button
     android:id="@+id/notify"       android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/notify_me"
app:layout_constraintBottom_toTopOf="@+id/update"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
   <Button
     android:id="@+id/update"       android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/update_me"
app:layout_constraintBottom_toTopOf="@+id/cancel"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/notify" />
   <Button
     android:id="@+id/cancel"       android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/cancel_me"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/update" />


</android.support.constraint.ConstraintLayout>
```

MainActivity.java package

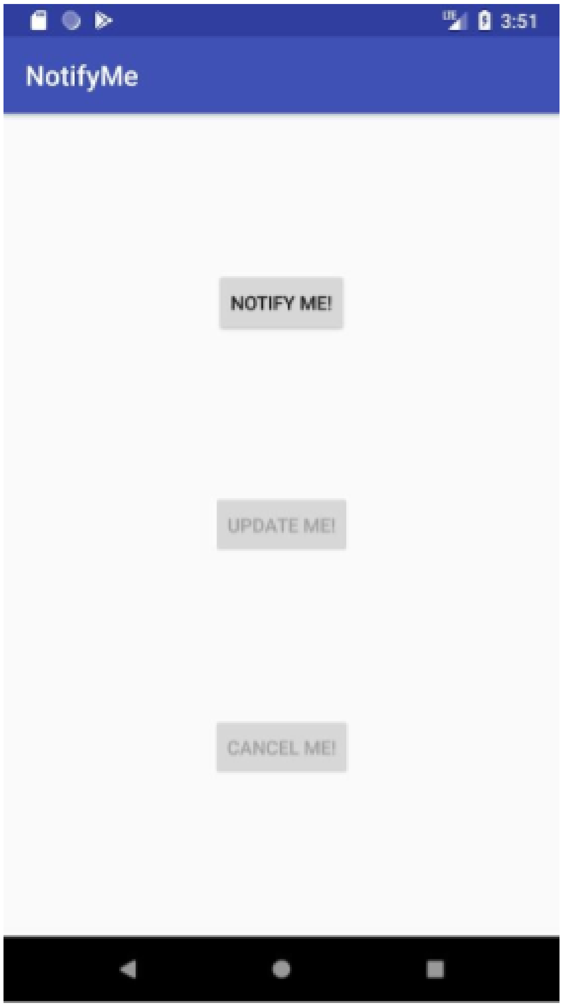
```
com.android.example.notifyme; import
android.app.NotificationChannel; import
android.app.NotificationManager; import
android.app.PendingIntent; import
android.content.BroadcastReceiver;
import android.content.Context; import
android.content.Intent; import
android.content.IntentFilter; import
android.graphics.Bitmap; import
android.graphics.BitmapFactory; import
android.graphics.Color; import
android.os.Bundle;
import android.support.v4.app.NotificationCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.Button;

/**
 * MainActivity for the Notify Me! app. Contains three buttons that deliver,  * update,
and cancel notification.
 */
public class MainActivity extends AppCompatActivity {

   // Constants for the notification actions buttons.
   private static final String ACTION_UPDATE_NOTIFICATION =
          "com.android.example.notifyme.ACTION_UPDATE_NOTIFICATION";
   // Notification channel ID.     private static final String
PRIMARY_CHANNEL_ID =
"primary_notification_channel";
   // Notification ID.     private static final int
NOTIFICATION_ID = 0;
    private Button button_notify;
private Button button_cancel;
private Button button_update;

   private NotificationManager mNotifyManager;
    private NotificationReceiver mReceiver = new NotificationReceiver();

   /**
*  Initializes the activity.
    *
*  @param savedInstanceState The current state data.
    */    @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_main);

       // Create the notification channel.
       createNotificationChannel();
    // Register the broadcast receiver to receive the update action from      // the
notification.
    registerReceiver(mReceiver,
        new IntentFilter(ACTION_UPDATE_NOTIFICATION));

    // Add onClick handlers to all the buttons.      button_notify =
findViewById(R.id.notify);
    button_notify.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {          //
Send the notification          sendNotification();
      }
    });
    button_update = (Button) findViewById(R.id.update);
button_update.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        // Update the notification.
        updateNotification();
      }
    });
    button_cancel = (Button) findViewById(R.id.cancel);
button_cancel.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        // Cancel the notification.
        cancelNotification();
      }
    });

    // Reset the button states. Enable only Notify button and disable
    // update and cancel buttons.      setNotificationButtonState(true, false,
false);
  }

  /**
* Unregisters the receiver when the app is being destroyed.
   */   @Override
  protected void onDestroy() {
unregisterReceiver(mReceiver);      super.onDestroy();
  }

  /**
```

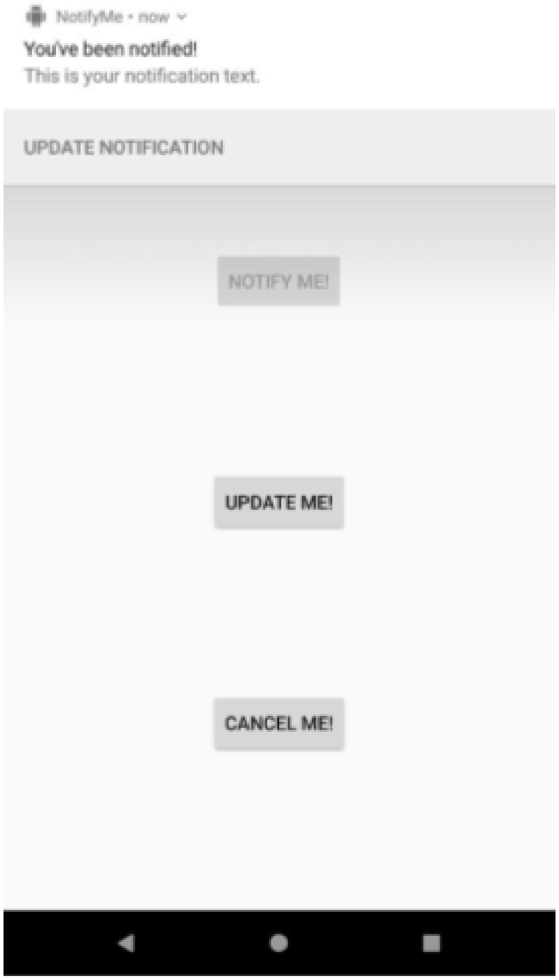
```
* Creates a Notification channel, for OREO and higher.
   */
  public void createNotificationChannel() {

     // Create a notification manager object.       mNotifyManager =
          (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
     // Notification channels are only available in OREO and higher.
     // So, add a check on SDK version.
     if (android.os.Build.VERSION.SDK_INT >=
                         android.os.Build.VERSION_CODES.O) {

        // Create the NotificationChannel with all the parameters.
        NotificationChannel notificationChannel = new NotificationChannel
             (PRIMARY_CHANNEL_ID,
                  getString(R.string.notification_channel_name),
                  NotificationManager.IMPORTANCE_HIGH);
         notificationChannel.enableLights(true);
notificationChannel.setLightColor(Color.RED);
notificationChannel.enableVibration(true);
notificationChannel.setDescription
             (getString(R.string.notification_channel_description));

        mNotifyManager.createNotificationChannel(notificationChannel);
     }
  }

  /**
* OnClick method for the "Notify Me!" button.
* Creates and delivers a simple notification.
   */
  public void sendNotification() {

     // Sets up the pending intent to update the notification.
     // Corresponds to a press of the Update Me! button.
     Intent updateIntent = new Intent(ACTION_UPDATE_NOTIFICATION);
     PendingIntent updatePendingIntent = PendingIntent.getBroadcast(this,
NOTIFICATION_ID, updateIntent, PendingIntent.FLAG_ONE_SHOT);
     // Build the notification with all of the parameters using helper
     // method.       NotificationCompat.Builder notifyBuilder =
getNotificationBuilder();

     // Add the action button using the pending intent.
     notifyBuilder.addAction(R.drawable.ic_update,
                  getString(R.string.update), updatePendingIntent);
        // Deliver the notification.         mNotifyManager.notify(NOTIFICATION_ID,
notifyBuilder.build());

        // Enable the update and cancel buttons but disables the "Notify
        // Me!" button.         setNotificationButtonState(false, true,
true);
    }

    /**
* Helper method that builds the notification.
     *
* @return NotificationCompat.Builder: notification build with all the
* parameters.
     */
    private NotificationCompat.Builder getNotificationBuilder() {

        // Set up the pending intent that is delivered when the notification
        // is clicked.
        Intent notificationIntent = new Intent(this, MainActivity.class);
        PendingIntent notificationPendingIntent = PendingIntent.getActivity
                (this, NOTIFICATION_ID, notificationIntent,
                        PendingIntent.FLAG_UPDATE_CURRENT);

        // Build the notification with all of the parameters.
        NotificationCompat.Builder notifyBuilder = new NotificationCompat
                .Builder(this, PRIMARY_CHANNEL_ID)
                .setContentTitle(getString(R.string.notification_title))
                .setContentText(getString(R.string.notification_text))
                .setSmallIcon(R.drawable.ic_android)
                .setAutoCancel(true).setContentIntent(notificationPendingIntent)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
.setDefaults(NotificationCompat.DEFAULT_ALL);         return notifyBuilder;
    }

    /**
* OnClick method for the "Update Me!" button. Updates the existing
* notification to show a picture.
     */
    public void updateNotification() {

        // Load the drawable resource into the a bitmap image.
        Bitmap androidImage = BitmapFactory
                .decodeResource(getResources(), R.drawable.mascot_1);
        // Build the notification with all of the parameters using helper
        // method.
        NotificationCompat.Builder notifyBuilder = getNotificationBuilder();
        // Update the notification style to BigPictureStyle.
        notifyBuilder.setStyle(new NotificationCompat.BigPictureStyle()
                .bigPicture(androidImage)
                .setBigContentTitle(getString(R.string.notification_updated)));
        // Deliver the notification.         mNotifyManager.notify(NOTIFICATION_ID,
notifyBuilder.build());

        // Disable the update button, leaving only the cancel button enabled.
setNotificationButtonState(false, false, true);
    }

    /**
* OnClick method for the "Cancel Me!" button. Cancels the notification.
     */
    public void cancelNotification() {
        // Cancel the notification.         mNotifyManager.cancel(NOTIFICATION_ID);

        // Reset the buttons.         setNotificationButtonState(true, false,
false);
    }

    /**
* Helper method to enable/disable the buttons.
     *
* @param isNotifyEnabled, boolean: true if notify button enabled
* @param isUpdateEnabled, boolean: true if update button enabled
* @param isCancelEnabled, boolean: true if cancel button enabled
     */
    void setNotificationButtonState(Boolean isNotifyEnabled, Boolean
isUpdateEnabled, Boolean isCancelEnabled) {
button_notify.setEnabled(isNotifyEnabled);         button_update.setEnabled(isUpdateEnabled);
button_cancel.setEnabled(isCancelEnabled);
    }

    /**
* The broadcast receiver class for notifications.
* Responds to the update notification pending intent action.
     */
    public class NotificationReceiver extends BroadcastReceiver {

        public NotificationReceiver() {
        }
        /**
* Receives the incoming broadcasts and responds accordingly.
         *
* @param context Context of the app when the broadcast is received.
* @param intent The broadcast intent containing the action.
         */
@Override
        public void onReceive(Context context, Intent intent) {
            // Update the notification.
            updateNotification();
        }
    } }
```

strings.xml

```
<resources>
    <string name="app_name">Notify Me!</string>
    <string name="notification_title">You\'ve been notified!</string>
    <string name="notification_text">This is your notification text.</string>
    <string name="update">Update Notification</string>
    <string name="notify_me">Notify Me!</string>
    <string name="update_me">Update Me!</string>
    <string name="cancel_me">Cancel Me!</string>
    <string name="notification_updated">Notification Updated!</string>
    <string name="notification_channel_name">Mascot Notification</string>
    <string name="notification_channel_description">Notification from Mascot</string>
</resources>
```

Output:





You've been notified!

This is your notification text.

Practical: 13

Practical: Get and Save User Preferences

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"     android:padding="16dp"
tools:context=".MainActivity">

  <TextView
      android:id="@+id/count_textview"
android:layout_width="0dp"        android:layout_height="0dp"
      android:background="@color/default_background"
android:gravity="center"        android:text="@string/default_count"
android:textColor="@android:color/white"        android:textSize="112sp"
      app:layout_constraintBottom_toTopOf="@+id/guideline_upper"
app:layout_constraintDimensionRatio="1:1"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"/>

  <android.support.constraint.Guideline
android:id="@+id/guideline_upper"        android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:orientation="horizontal"
app:layout_constraintGuide_end="120dp"/>
   <Button
      android:id="@+id/black_background_button"
style="@style/AppTheme.Button.Colored"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@android:color/black"
android:onClick="changeBackground"
android:text="@string/black_button"
      app:layout_constraintBottom_toTopOf="@+id/guideline_lower"
app:layout_constraintEnd_toStartOf="@+id/red_background_button"
app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="@+id/guideline_upper"/>

  <Button
```

```
      android:id="@+id/red_background_button"
style="@style/AppTheme.Button.Colored"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@color/red_background"
android:onClick="changeBackground"        android:text="@string/red_button"
      app:layout_constraintBottom_toTopOf="@+id/guideline_lower"
app:layout_constraintEnd_toStartOf="@+id/blue_background_button"
app:layout_constraintStart_toEndOf="@+id/black_background_button"
app:layout_constraintTop_toTopOf="@+id/guideline_upper"/>
   <Button
      android:id="@+id/blue_background_button"
style="@style/AppTheme.Button.Colored"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@color/blue_background"
android:onClick="changeBackground"        android:text="@string/blue_button"
      app:layout_constraintBottom_toTopOf="@+id/guideline_lower"
app:layout_constraintEnd_toStartOf="@+id/green_background_button"
app:layout_constraintStart_toEndOf="@+id/red_background_button"
app:layout_constraintTop_toTopOf="@+id/guideline_upper"/>
   <Button
      android:id="@+id/green_background_button"
style="@style/AppTheme.Button.Colored"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="@color/green_background"
android:onClick="changeBackground"
android:text="@string/green_button"
      app:layout_constraintBottom_toTopOf="@+id/guideline_lower"
app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toEndOf="@+id/blue_background_button"
app:layout_constraintTop_toTopOf="@+id/guideline_upper"/>

  <android.support.constraint.Guideline
android:id="@+id/guideline_lower"        android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:orientation="horizontal"
app:layout_constraintGuide_end="56dp"/>
   <Button
      android:id="@+id/count_button"
style="@style/AppTheme.Button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="16dp"
android:layout_marginRight="16dp"        android:onClick="countUp"
```

```
android:text="@string/count_button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/reset_button"
app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/guideline_lower"/>

   <Button
      android:id="@+id/reset_button"        style="@style/AppTheme.Button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"        android:onClick="reset"
android:text="@string/reset_button"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/count_button"
app:layout_constraintTop_toBottomOf="@+id/guideline_lower"/>

</android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
/*
*         Copyright (C) 2016 Google Inc.
 *
*         Licensed under the Apache License, Version 2.0 (the "License");  * you may not
          use this file except in compliance with the License.
*         You may obtain a copy of the License at
 *
*         http://www.apache.org/licenses/LICENSE-2.0
 *
*         Unless required by applicable law or agreed to in writing, software
*         distributed under the License is distributed on an "AS IS" BASIS,  * WITHOUT
          WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  *
          See the License for the specific language governing permissions and  *
          limitations under the License.
 */
package com.example.android.hellosharedprefs;
 import android.content.SharedPreferences; import
android.graphics.drawable.ColorDrawable; import
android.support.v4.content.ContextCompat; import
android.support.v7.app.AppCompatActivity; import
android.os.Bundle; import android.view.View; import
android.widget.TextView;

/**
```

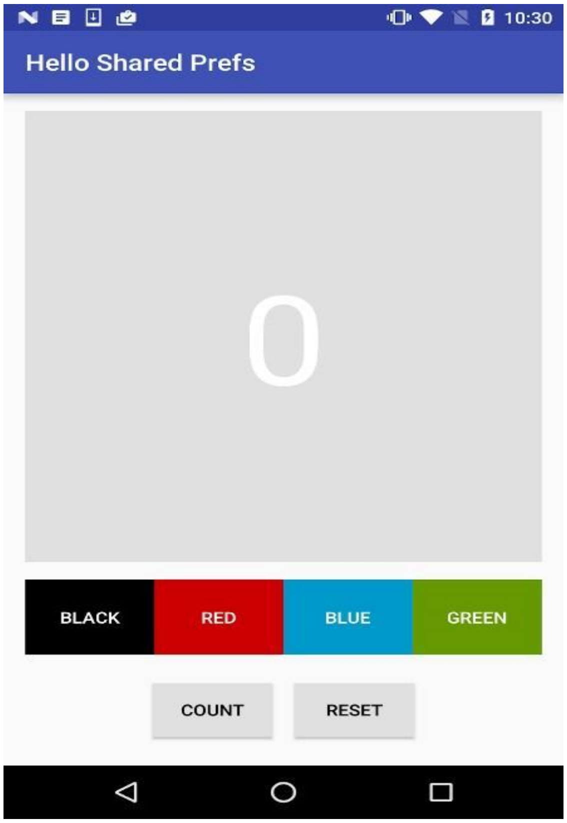
```
*         HelloSharedPrefs is an adaptation of the HelloToast app from chapter 1.
*         It includes:
*         - Buttons for changing the background color.
*         - Maintenance of instance state.   * - Themes and styles.
*         - Read and write shared preferences for the current count and the color.
 *  <p>
 * This is the solution code for HelloSharedPrefs.
 */
public class MainActivity extends AppCompatActivity {
   // Current count     private int
mCount = 0;     // Current
background color     private int
mColor;
   // Text view to display both count and color     private
TextView mShowCountTextView;

   // Key for current count
   private final String COUNT_KEY = "count";
   // Key for current color
   private final String COLOR_KEY = "color";

   // Shared preferences object
   private SharedPreferences mPreferences;

   // Name of shared preferences file     private
String sharedPrefFile =
         "com.example.android.hellosharedprefs";


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

       // Initialize views, color, preferences
       mShowCountTextView = findViewById(R.id.count_textview);
       mColor = ContextCompat.getColor(this,
             R.color.default_background);
       mPreferences = getSharedPreferences(sharedPrefFile, MODE_PRIVATE);

       // Restore preferences
       mCount = mPreferences.getInt(COUNT_KEY, 0);
mShowCountTextView.setText(String.format("%s", mCount));        mColor =
mPreferences.getInt(COLOR_KEY, mColor);
mShowCountTextView.setBackgroundColor(mColor);
```

```
    }

    /**
*   Handles the onClick for the background color buttons.  Gets background
*   color of the button that was clicked, and sets the TextView background
*   to that color.
     *
*   @param view The view (Button) that was clicked.
     */
    public void changeBackground(View view) {
        int color = ((ColorDrawable) view.getBackground()).getColor();
mShowCountTextView.setBackgroundColor(color);          mColor = color;
    }

    /**
*   Handles the onClick for the Count button.  Increments the value of the
*   mCount global and updates the TextView.
     *
*   @param view The view (Button) that was clicked.
     */
    public void countUp(View view) {
mCount++;
        mShowCountTextView.setText(String.format("%s", mCount));
    }

    /**
*   Handles the onClick for the Reset button.  Resets the global count and       *
    background variables to the defaults and resets the views to those
*   default values.
     *
*   @param view The view (Button) that was clicked.
     */
    public void reset(View view) {
        // Reset count         mCount
= 0;
        mShowCountTextView.setText(String.format("%s", mCount));

        // Reset color
        mColor = ContextCompat.getColor(this,
R.color.default_background);
mShowCountTextView.setBackgroundColor(mColor);

        // Clear preferences
        SharedPreferences.Editor preferencesEditor = mPreferences.edit();
preferencesEditor.clear();         preferencesEditor.apply();
    }

    /**
     * Callback for activity pause.  Shared preferences are saved here.
     */     @Override
    protected void onPause() {         super.onPause();

        SharedPreferences.Editor preferencesEditor = mPreferences.edit();
preferencesEditor.putInt(COUNT_KEY, mCount);
preferencesEditor.putInt(COLOR_KEY, mColor);         preferencesEditor.apply();
    } }
```

strings.xml

```
<resources>
    <!-- Title of app -->
    <string name="app_name">HelloSharedPrefs</string>

    <!-- Button label for blue button -->
    <string name="blue_button">Blue</string>

    <!-- Button label for green button -->
    <string name="green_button">Green</string>

    <!-- Button label for red button -->
    <string name="red_button">Red</string>

    <!-- Button label for black button -->
    <string name="black_button">Black</string>

    <!-- Default string for count textview -->
    <string name="default_count">0</string>

    <!-- Button label for Count button -->
    <string name="count_button">Count</string>

    <!-- Button label for Reset button -->
    <string name="reset_button">Reset</string>
</resources>
```

Output:



Practical: 14

Practical: Using location service get the current location and display in TextView

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.currentlocationdemo">

   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"       />
<uses-permission   android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
   <application
      android:allowBackup="true"        android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
      android:theme="@style/Theme.CurrentLocationDemo">
      <activity android:name=".MainActivity">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>

   <meta-data
      android:name="com.google.android.geo.API_KEY"
      android:value="AIzaSyBZ2tydt2CWrBTVfisn8GO50MnTX615gE0" />

</manifest>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
```

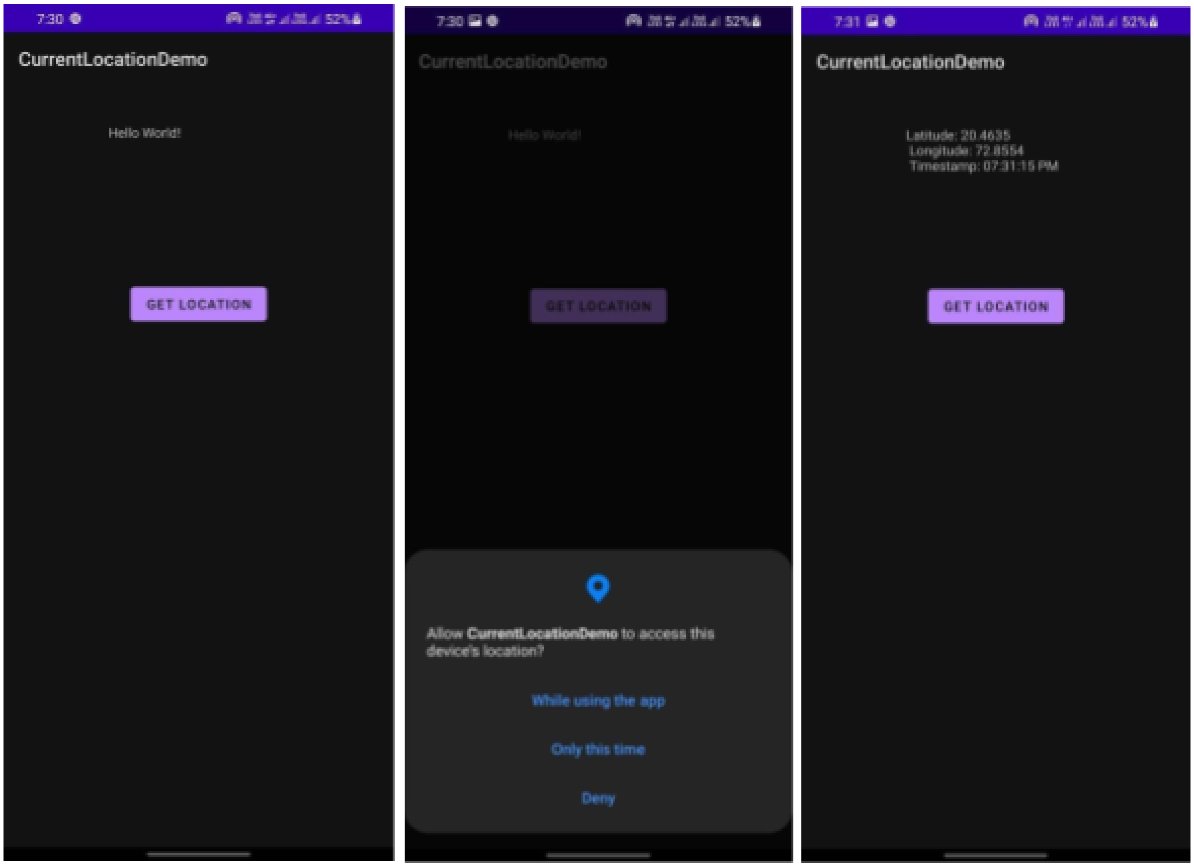
```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"    android:layout_height="match_parent"
tools:context=".MainActivity">

  <TextView
      android:id="@+id/textView"
android:layout_width="188dp"
android:layout_height="101dp"      android:text="Hello
World!"
      app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.058" />
   <Button
      android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"      android:text="Get
Location"
      app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.275" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.currentlocationdemo;

import androidx.annotation.NonNull; import
androidx.appcompat.app.AppCompatActivity; import
androidx.core.app.ActivityCompat;

import android.Manifest; import
android.content.pm.PackageManager; import
android.location.Address; import
android.location.Geocoder; import
android.location.Location; import
android.os.Bundle; import android.util.Log;
```

```
import android.view.View; import
android.widget.Button; import
android.widget.TextView; import
android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices; import
com.google.android.gms.maps.model.LatLng; import
com.google.android.gms.tasks.OnSuccessListener;
import java.io.IOException;
import java.util.List; import
java.util.Locale;

public class MainActivity extends AppCompatActivity {

   Location mLastLocation;
   TextView t;
   Button b;
   FusedLocationProviderClient mFusedLocationClient;
   Geocoder geocoder;
   LatLng sydney = new LatLng(-34, 151);
   LatLng currentLoc;
   List<Address> addresses;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);        setContentView(R.layout.activity_main);
       b = findViewById(R.id.button);        t =
findViewById(R.id.textView);
      b.setOnClickListener(new View.OnClickListener() {
@Override
         public void onClick(View v) {              getLocation();
         }
      });

      mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);     }
   private void getLocation() {
      if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
             != PackageManager.PERMISSION_GRANTED) {
         ActivityCompat.requestPermissions(this, new String[]
                {Manifest.permission.ACCESS_FINE_LOCATION}, 1);
      } else {
            mFusedLocationClient.getLastLocation().addOnSuccessListener(
new OnSuccessListener<Location>() {
                        @Override
                        public void onSuccess(Location location) {
if (location != null) {                         mLastLocation = location;
                                t.setText(
                                                                          getString(R.string.location_text,
mLastLocation.getLatitude(),                                               mLastLocation.getLongitude(),
mLastLocation.getTime()));                                                           currentLoc  =  new
LatLng(mLastLocation.getLatitude(), mLastLocation.getLongitude());                              } else {
                                t.setText("No Location");
                            }
                        }
                    });

            /*geocoder = new Geocoder(this, Locale.getDefault());             try {
                addresses = geocoder.getFromLocation(sydney.latitude, sydney.longitude, 1);
                //List<Address> addresses = geocoder.getFromLocationName("731 Market
St, San Francisco, CA 94103", 1);
                Address firstAddress = addresses.get(0);                 double
latitude = firstAddress.getLatitude();                 double longitude =
firstAddress.getLongitude();
                t.setText(firstAddress.getAddressLine(0) + "\n" + "Latitude:" + latitude +"\n" +
"Longitude:" + longitude);             } catch (IOException e) {
                e.printStackTrace();
            }*/

            /*if(addresses!=null){
                Address a = addresses.get(0);                 StringBuilder s
= new StringBuilder();
                s.append(a.getAddressLine(0));
                t.setText(s.toString());
            }*/
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           @NonNull String[] permissions, @NonNull int[] grantResults) {
switch (requestCode) {             case 1:
                // If the permission is granted, get the location,
                // otherwise, show a Toast
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
getLocation();
              } else {
                  Toast.makeText(this, "Location Permission Denied!",
Toast.LENGTH_SHORT).sho
w();                }
break;
        }
    } }
```

Output:



Hello World!

Hello World!

Allow **CurrentLocationDemo** to access this device's location?

Latitude: 20.4635
Longitude: 72.8554
Timestamp: 07:31:15 PM

Practical: 15

Practical: Display the use of animations

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"     android:layout_height="match_parent"
tools:context=".MainActivity">

   <com.example.practical4.MyCircle        android:layout_width="match_parent"
android:layout_height="match_parent"/>

</RelativeLayout>
```

MainActivity.java

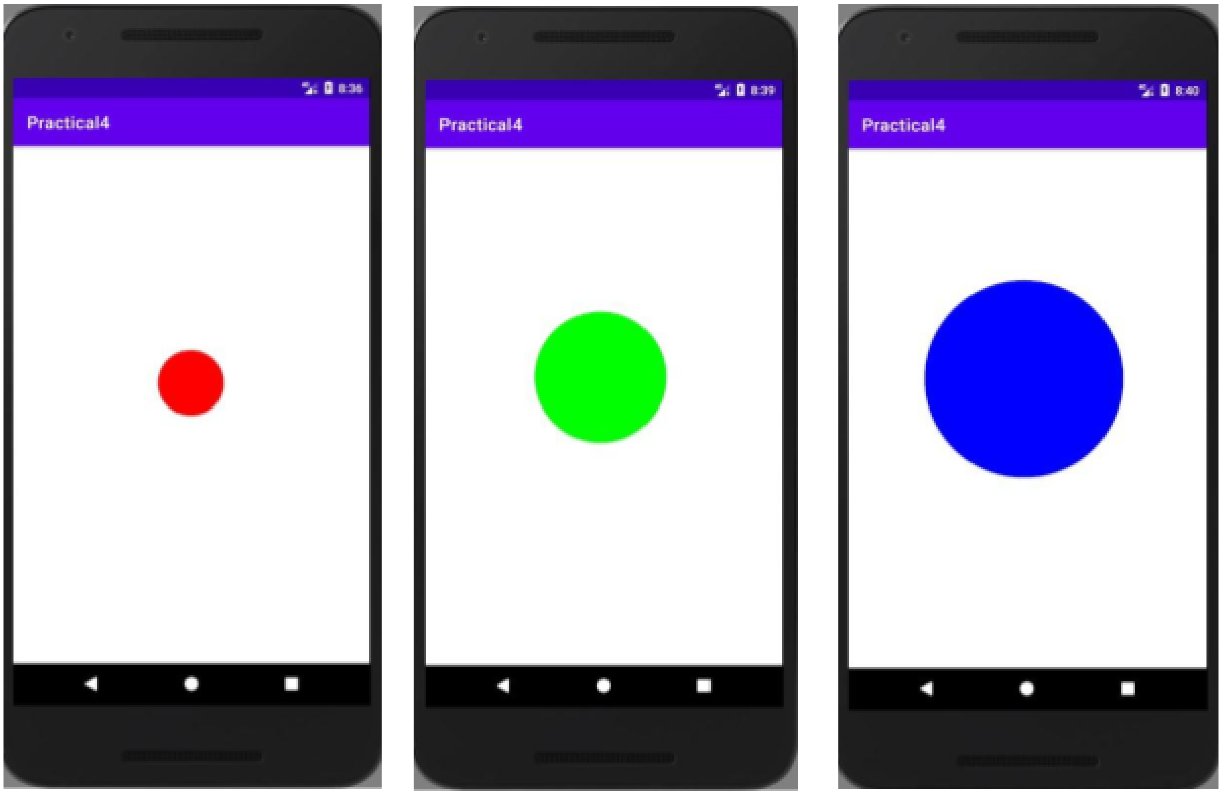
```
package com.example.practical4;

import androidx.appcompat.app.AppCompatActivity;
 import android.animation.AnimatorInflater;
import android.animation.AnimatorSet; import
android.graphics.Color; import
android.graphics.Paint;
import android.graphics.drawable.ShapeDrawable;
import android.graphics.drawable.shapes.Shape;
import android.os.Bundle; import android.view.View;
import android.view.ViewGroup; import
android.view.animation.Animation; import
android.view.animation.AnimationUtils;
 public class MainActivity extends AppCompatActivity {


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

MyCircle.java package

```
com.example.practical4;

import android.animation.AnimatorSet;
import android.animation.ArgbEvaluator;
import android.animation.ObjectAnimator;
import android.animation.ValueAnimator;
import android.content.Context; import
android.graphics.Canvas; import
android.graphics.Color; import
android.graphics.Paint;
import android.graphics.drawable.ColorDrawable;
import android.util.AttributeSet; import
android.view.MotionEvent; import
android.view.View;
import android.view.animation.LinearInterpolator;

import androidx.interpolator.view.animation.LinearOutSlowInInterpolator;

public class MyCircle extends View {
private float mRadius;     private Paint
mPaint = new Paint();     private float mX;
private float mY;
   private static final int ANIMATION_DURATION = 4000;     private
static final long ANIMATION_DELAY = 1000;     private AnimatorSet
mPulseAnimatorSet = new AnimatorSet();

   public MyCircle(Context context) {        super(context);
      mPaint.setColor(Color.RED);
   }
         public  MyCircle(Context  context,  AttributeSet  attrs)  {
super(context, attrs);        mPaint.setColor(Color.RED);
   }
   public void setRadius(float radius) {
mRadius = radius;        invalidate();
   }

   @Override
   public boolean onTouchEvent(MotionEvent event) {        if
(event.getActionMasked() == MotionEvent.ACTION_DOWN) {
mX = event.getX();            mY = event.getY();

         if(mPulseAnimatorSet != null && mPulseAnimatorSet.isRunning()) {
mPulseAnimatorSet.cancel();
         }
         mPulseAnimatorSet.start();
      }
      return super.onTouchEvent(event);
   }

   @Override
```

```
    protected void onDraw(Canvas canvas) {         super.onDraw(canvas);
        canvas.drawCircle(mX, mY, mRadius, mPaint);
    }


    @Override
    public void onSizeChanged(int w, int h, int oldw, int oldh) {

        ObjectAnimator growAnimator = ObjectAnimator.ofFloat(this,
                "radius", 0, 100);
        growAnimator.setDuration(ANIMATION_DURATION);

        ObjectAnimator rotateAnimator = ObjectAnimator.ofFloat(this,
MyCircle.ROTATION, 0f,360f);
        rotateAnimator.setDuration(ANIMATION_DURATION);

        ObjectAnimator colorAnimator = ObjectAnimator.ofInt(mPaint, "color",
Color.GREEN, Color.GREEN);
        colorAnimator.setDuration(ANIMATION_DURATION);
colorAnimator.setStartDelay(ANIMATION_DELAY);

        ObjectAnimator growAnimator1 = ObjectAnimator.ofFloat(this,
                "radius", 100, 200);
        growAnimator1.setDuration(ANIMATION_DURATION);

        ObjectAnimator rotateAnimator1 = ObjectAnimator.ofFloat(this,
MyCircle.ROTATION, 0f,360f);
        rotateAnimator1.setDuration(ANIMATION_DURATION);

        ObjectAnimator colorAnimator1 = ObjectAnimator.ofInt(mPaint, "color",
Color.BLUE, Color.BLUE);
        colorAnimator1.setDuration(ANIMATION_DURATION);
colorAnimator1.setStartDelay(ANIMATION_DELAY);

        ObjectAnimator growAnimator2 = ObjectAnimator.ofFloat(this,
                "radius", 200, 300);
        growAnimator2.setDuration(ANIMATION_DURATION);

        ObjectAnimator rotateAnimator2 = ObjectAnimator.ofFloat(this,
MyCircle.ROTATION, 0f,360f);
        rotateAnimator2.setDuration(ANIMATION_DURATION);
         mPulseAnimatorSet.play(growAnimator).before(rotateAnimator);
mPulseAnimatorSet.play(growAnimator1).after(rotateAnimator);
mPulseAnimatorSet.play(colorAnimator).with(growAnimator1);
mPulseAnimatorSet.play(rotateAnimator1).after(growAnimator1);
mPulseAnimatorSet.play(growAnimator2).after(rotateAnimator1);
mPulseAnimatorSet.play(growAnimator2).with(colorAnimator1);
mPulseAnimatorSet.play(rotateAnimator2).after(growAnimator2);           }

}
```

Output:


8:36
Practical4
8:39
Practical4
8:40
Practical4