CIS 470
Mobile App Development

Lecture 4

Wenbing Zhao
Department of Electrical Engineering and Computer Science
Cleveland State University
wenbing@ieee.org

Intents

- Returning data from an activity
- Passing data to an activity

Returning data from an activity

- Modify the UsingIntent app to see how this is done
- Replace secondactivity.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    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: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="com.username.usingintent.SecondActivity">
```

Returning data from an activity

- Replace secondactivity.xml:

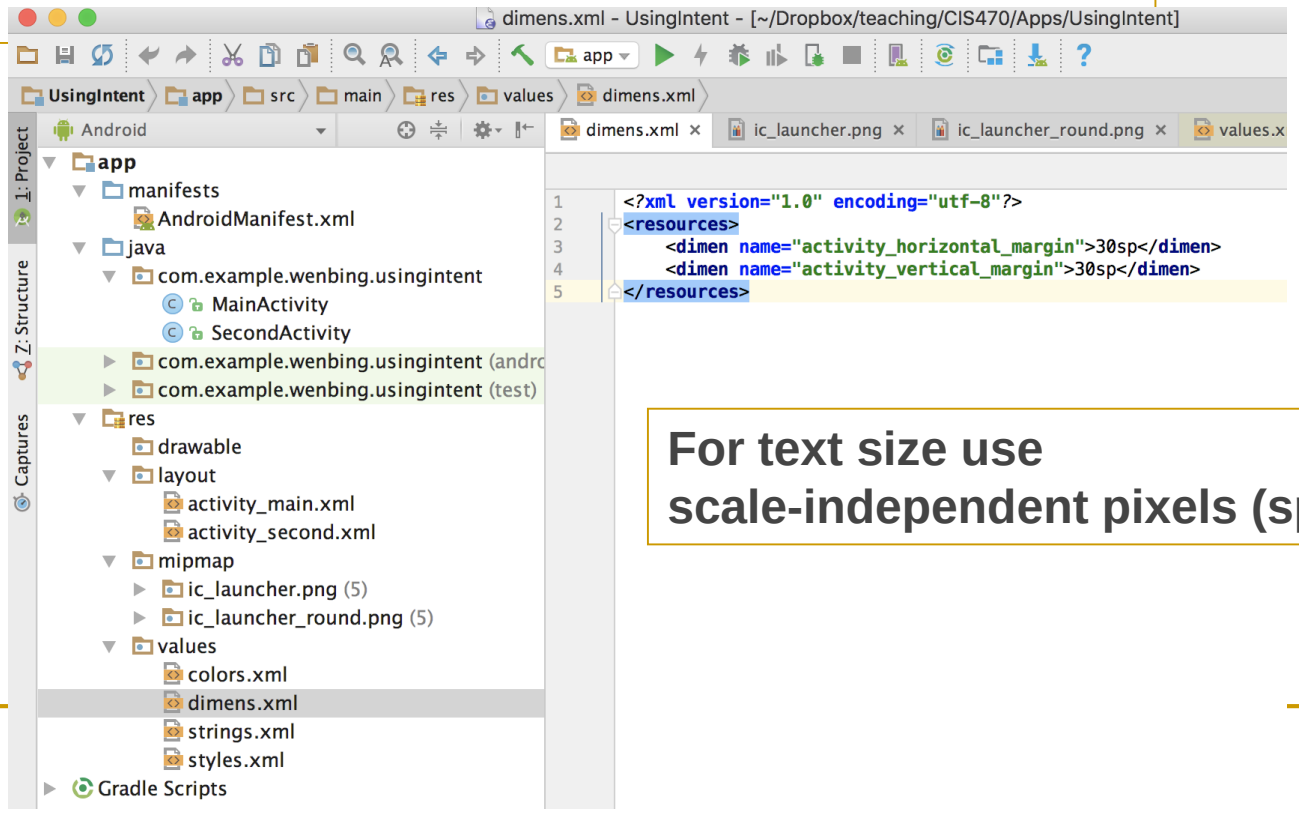
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is the Second Activity!"
    android:id="@+id/textView2" />
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Please enter your name"
   android:id="@+id/textView3" />
```

```
   <EditText
      android:layout_width="match_parent”
      android:layout_height="wrap_content"
      android:id="@+id/txtUsername" />
   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="OK"
      android:onClick="onClick"
      android:id="@+id/button2" />
</LinearLayout>
```

Returning data from an activity

- Create a dimen.xml file to define dimen values:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <dimen name="activity_horizontal_margin">30sp</dimen>
   <dimen name="activity_vertical_margin">30sp</dimen>
</resources>
```



For text size use scale-independent pixels (sp)

Returning data from an activity

- Add the **bolded** statements in the following code to SecondActivity.java:

```
package com.example.wenbing.usingintent;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.content.Intent;
import android.net.Uri;

public class SecondActivity extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
    }
```

Returning data from an activity

- Add the **bolded** statements in the following code to SecondActivity.java:

```
public class SecondActivity extends Activity {
    .....
    public void onClick(View view) {
        Intent data = new Intent();
//---get the EditText view---
        EditText txt_username = (EditText)findViewById(R.id.txtUsername);
//---set the data to pass back---
        data.setData(Uri.parse( txt_username.getText().toString()));
        setResult(RESULT_OK, data);
//---closes the activity---
        finish();
    }
}
```

For an activity to return a value to the calling activity, you use an Intent object to send data back via the setData() method

The setResult() method sets a result code (either RESULT_OK or RESULT_CANCELLED) and the data (an Intent object) to be returned back to the calling activity

Returning data from an activity

- Add (or replace with) the bolded statements in the following code to the MainActivity.java file:

```
import android.view.View;
import android.widget.Toast;
public class MainActivity extends Activity {
    int request_Code = 1;
    .....
    public void onClick(View view) {
        startActivityForResult(new Intent("com.username.usingintent.SecondActivity"),request_Code);
    }
    public void onActivityResult(int requestCode, int resultCode, Intent data)
    {
        if (requestCode == request_Code) {
            if (resultCode == RESULT_OK) {
                Toast.makeText(this,data.getData().toString(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

The request code is an integer value that identifies an activity you are calling (cannot be -1)

When an activity returns, you need this request code to determine which activity is actually returned

A toast is a view containing a quick little message for the user

To retrieve the data set using thesetData()method, use the getData() method

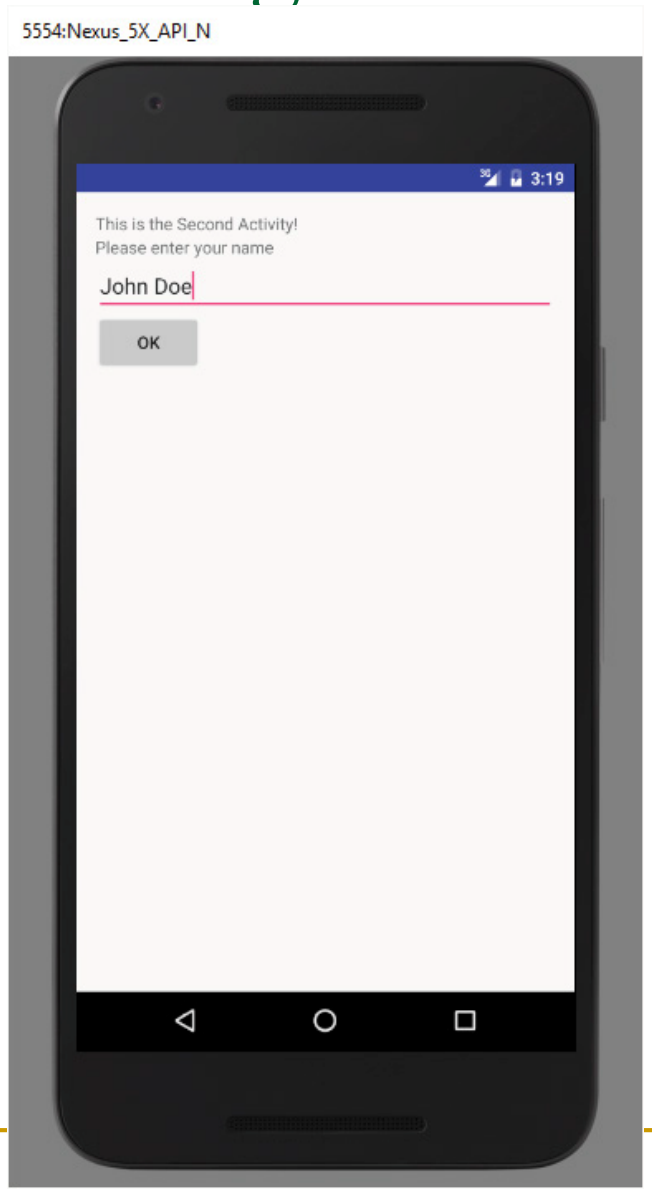
Toast

public class Toast extends Object
java.lang.Object
↳android.widget.Toast

- A toast is a view containing a quick little message for the user
- When the view is shown to the user, appears as a floating view over the application
- The idea is to be as unobtrusive as possible, while still showing the user the information you want them to see Two examples are the volume control, and the brief message saying that your settings have been saved

int	LENGTH_LONGShow the view or text notification for a long period of time.
int	LENGTH_SHORTShow the view or text notification for a short period of time.

Returning data from an activity



Passing data to an activity

- Create a new project and name it **PassingData**
- Add/replace with the bolded statements in the following code to the activity_main.xml file:

Create a dimen.xml file to define dimen values

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
   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: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="com.username.passingdata.MainActivity">
   <Button
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Click to go to Second Activity"
       android:id="@+id/button"
       android:onClick="onClick"/>
</LinearLayout>
```

Replace old code with this!

Passing data to an activity

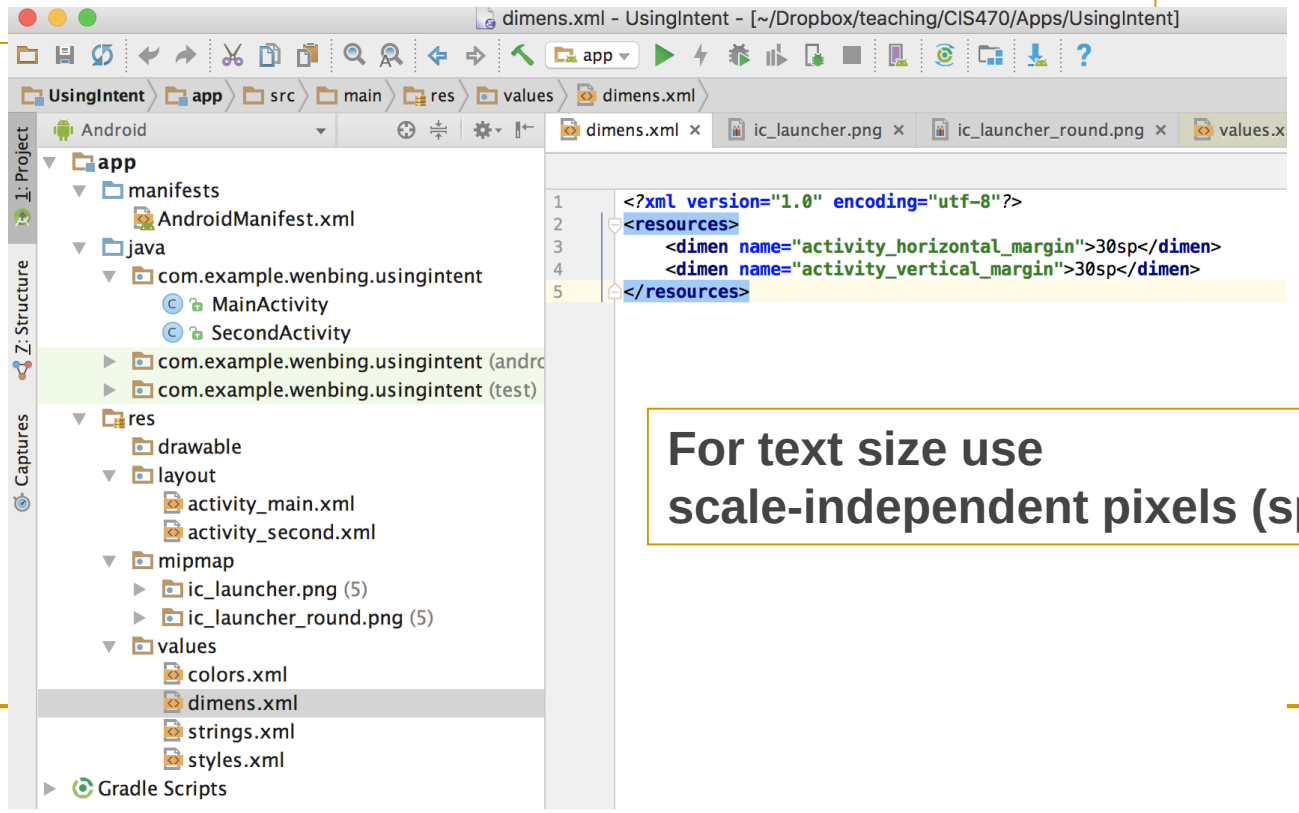
- Add a new XML file to the res/layout folder and name it activity_second.xml. Populate it as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
    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: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="com.username.passingdata.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to the Second Activity"
        android:id="@+id/textView" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click to go to Main Activity"
        android:id="@+id/button"
        android:onClick="onClick"/>
</LinearLayout>
```

Passing data to an activity

- Create a dimen.xml file to define dimen values:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <dimen name="activity_horizontal_margin">30sp</dimen>
   <dimen name="activity_vertical_margin">30sp</dimen>
</resources>
```



For text size use scale-independent pixels (sp)

Passing data to an activity

- Add a new Class file to the package and name it SecondActivity. Populate the SecondActivity.java file as follows:

```
package com.username.passingdata;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
```

```
public class SecondActvity extends Activity {
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_second);
   //---get the data passed in using getStringExtra()---
      Toast.makeText(this,getIntent().getStringExtra("str1"),
           Toast.LENGTH_SHORT).show();
   //---get the data passed in using getIntExtra()---
      Toast.makeText(this,Integer.toString(
            getIntent().getIntExtra("age1", 0)),
            Toast.LENGTH_SHORT).show();
   //---get the Bundle object passed in---
       Bundle bundle = getIntent().getExtras();
   //---get the data using the getString()---
       Toast.makeText(this, bundle.getString("str2"),
             Toast.LENGTH_SHORT).show();
   //---get the data using the getInt() method---
       Toast.makeText(this,Integer.toString(bundle.getInt("age2")),
             Toast.LENGTH_SHORT).show();
}
```

To obtain the data sent using the Intent object, you first obtain the Intent object using the getIntent() method. Then, call its getStringExtra() method to get the string value set using The putExtra() method

use getInt() method to retrieve an integer value

Passing data to an activity

- Add a new Class file to the package and name it SecondActivity. Populate the SecondActivity.java file as follows:

```
    public void onClick(View view) {
        //---use an Intent object to return data---
        Intent i = new Intent();
        //---use the putExtra() method to return some value---
        i.putExtra("age3", 45);
        //---use the setData() method to return some value---
        i.setData(Uri.parse("Something passed back to main activity"));
        //---set the result with OK and the Intent object---
        setResult(RESULT_OK, i);
        //---destroy the current activity---
        finish();
    }
}
```

Passing data to an activity

- Add the bolded statements from the following code to the AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.username.passingdata">
  <application
  ...
    <activity android:name=".SecondActvity">
      <intent-filter>
        <action android:name="com.username.passingdata.SecondActivity" />
        <category android:name="android.intent.category.DEFAULT" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

Passing data to an activity

- Add the bolded statements from the following code to the MainActivity.java file:

```
import android.content.Intent;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
```

Passing data to an activity

- Add the bolded statements from the following code to the MainActivity.java file:

```
public void onClick(View view) {
        Intent i = new
            Intent("com.username.passingdata.SecondActivity");
        //---use putExtra() to add new name/value pairs---
        i.putExtra("str1", "This is a string");
        i.putExtra("age1", 25);
        //---use a Bundle object to add new name/values pairs---
        Bundle extras = new Bundle();
        extras.putString("str2", "This is another string");
        extras.putInt("age2", 35);
        //---attach the Bundle object to the Intent object---
        i.putExtras(extras);
        //---start the activity to get a result back---
        startActivityForResult(i, 1);
}
```

putExtra() method of an Intent object to add a name/value pair

You can also create a Bundle object and then attach it using the putExtras()method Bundle object contains a set of name/value pairs.

Passing data to an activity

- Add the bolded statements from the following code to the MainActivity.java file:

```
    public void onActivityResult(int requestCode, int resultCode, Intent data)
    {
        //---check if the request code is 1---
        if (requestCode == 1) {
            //---if the result is OK---
            if (resultCode == RESULT_OK) {
                //---get the result using getIntExtra()---
                Toast.makeText(this, Integer.toString(
                    data.getIntExtra("age3", 0)), Toast.LENGTH_SHORT).show();
                //---get the result using getData()---
                Toast.makeText(this, data.getData().toString(),
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

For the integer value, use the getIntExtra() method

To retrieve the Bundle object, use the getExtras() method

To get the individual name/value pairs, use the appropriate method. For the string value, use the getString() method

Challenge Task#2

- Develop an contact management app (without persistency for now)
  - The main activity would start with no contact, but it has a button where the user could use to add a new contact
  - On clicking the button in the main activity, it goes to the second activity, where the user could add the contact information such as name, phone number, email, etc.
  - After entering a new contact, the user could click the submit button in the second activity, which the app will go back to the main activity
  - When there is at least one contact, the main activity could display a list of contacts identified by the full name of each contact
  - In the main activity, a user could click each row of the contact, a third activity would open to display the contact details