Mobile Application Development

Lesson 6

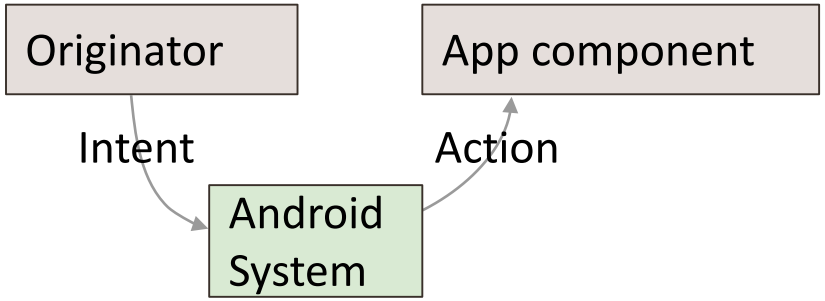
Dr. Syed Asim Jalal
Department of Computer Science
University of Peshawar

Intents and Passing Data to Activities

What is an intent?

- An `Intent` is a description of an operation to be performed.
- **Android Intent** is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.
- An `Intent` is an object used to request an action from another app component via the Android system.

Intents allow us to create an object that can be “given” to another component (read: Activity), who can then respond upon receiving it.



About intents

All Android activities are started or activated with an *intent*. Intents are message objects that make a request to the Android runtime to start an activity or other app component in your app or in some other app. You don't start those activities yourself;

When your app is first started from the device home screen, the Android runtime sends an intent to your app to start your app's main activity (the one defined with the MAIN action and the LAUNCHER category in the Android Manifest). To start other activities in your app, or request that actions be performed by some other activity available on the device, you build your own intents with the Intent class and call the startActivity() method to send that intent.

In addition to starting activities, intents are also used to pass data between activities. When you create an intent to start a new activity, you can include information about the data you want that new activity to operate on. So, for example, an email activity that displays a list of messages can send an intent to the activity that displays that message. The display activity needs data about the message to display, and you can include that data in the intent.

What can intents do?

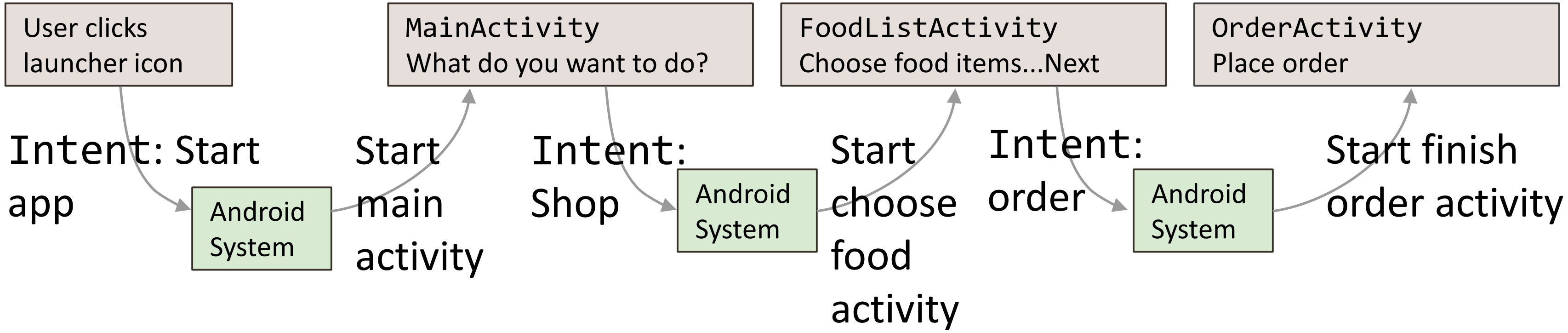
- Start an `Activity`
    - A button click starts a new `Activity` for text entry
    - Clicking Share opens an app that allows you to post a photo
- Start an `Service`
    - Initiate downloading a file in the background
- Deliver `Broadcast`
    - The system informs other apps that the phone is now charging

Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.

How Activities Run

- All `Activity` instances are managed by the Android runtime
- Started by an "`Intent`", a message to the Android runtime to open another activity
- For example, a food order application mention below


User clicks launcher icon
`Intent`: Start app
Android System
Start main activity
`MainActivity` What do you want to do?
`Intent`: Shop
Android System
Start choose food activity
`FoodListActivity` Choose food items...Next
`Intent`: order
Android System
Start finish order activity
`OrderActivity` Place order

Intent Types

1. **Explicit Intent**
2. **Implicit Intent**

Explicit Intent

- `Explicit Intent specifies the component to invoke. Intent provides the external class to be invoked.`
- Starts a specific `Activity`
  - Request an activity to update a profile picture in an app
  - Request an activity to show products in a shopping app

Start an Activity with an explicit intent

The most basic kind of Intent is an Intent sent to a specific Activity, such as for telling that Activity to open.

To start a specific `Activity`, use an explicit `Intent`

```
//                           context,           target
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
startActivity(intent);
```

1. Create an `Intent`

```
Intent intent = new Intent(this, ActivityName.class);
```

2. Use the `Intent` to start the `Activity`

```
startActivity(intent);
```

Implicit intents

- **Implicit Intent** doesn't specify the component.
- Asks system to **find** an `Activity` that can handle a request
    - Find an app and make a phone call with that app
    - Find a map app and display a location on that app
    - Find an app to view the webpage.

Start an Activity with implicit intent

To ask Android to find an `Activity` to handle your request, use an implicit `Intent`

1. Create an `Intent`:

```
Intent intent = new Intent( action, uri );
```

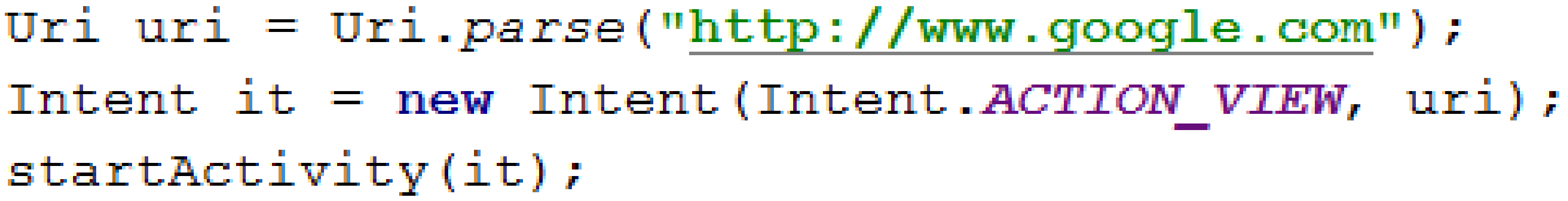
2. Use the `Intent` to start the `Activity`

```
startActivity(intent);
```

Implicit Intents - Examples

In this Intent we want the android operating system to pass our request to another other application to fulfill our request of displaying our URL as webpage. The Android will look for what browsers available on the phone.

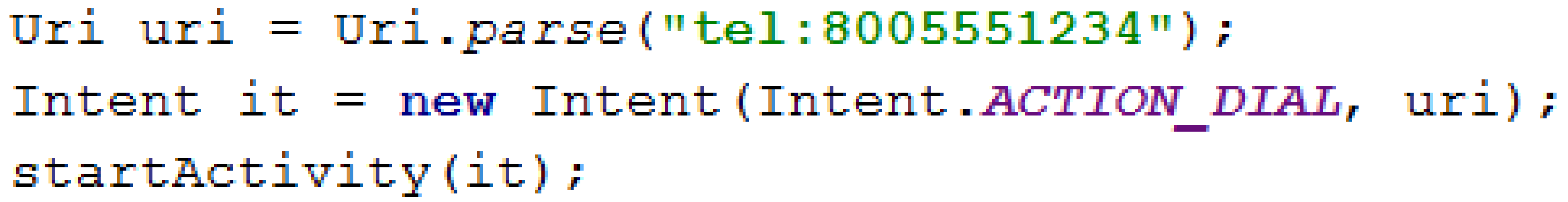
Show a web page



```
Uri uri = Uri.parse("http://www.google.com");
Intent it = new Intent(Intent.ACTION_VIEW, uri);
startActivity(it);
```

In this Intent we provide a phone number to the URI and request Android to look for any application that will fulfil our request of calling the phone number.

Dial a phone number



```
Uri uri = Uri.parse("tel:8005551234");
Intent it = new Intent(Intent.ACTION_DIAL, uri);
startActivity(it);
```

Intent types

There are two types of intents in Android:

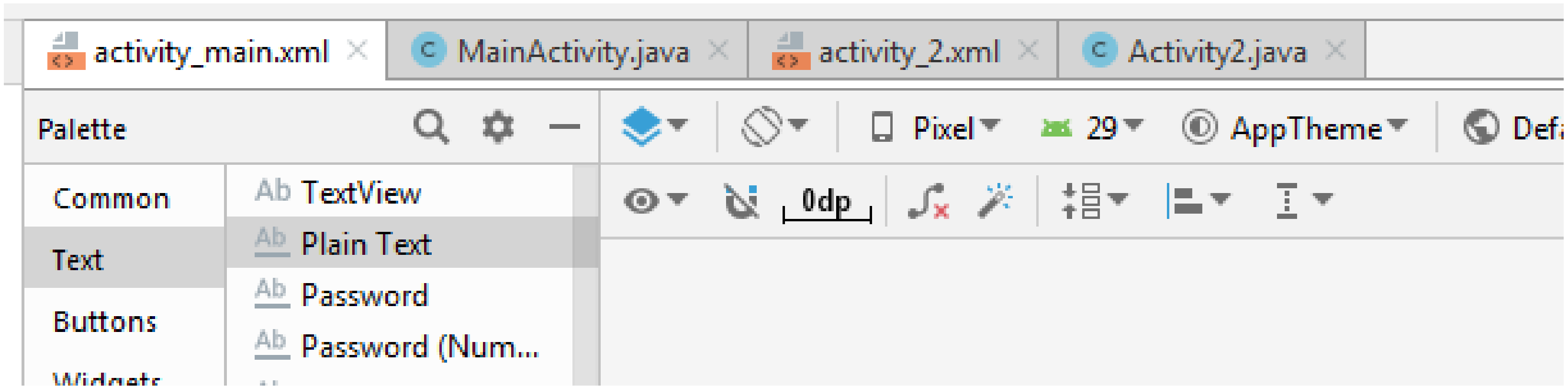
- *Explicit intents* specify the receiving activity (or other component) by that activity's fully-qualified class name. Use an explicit intent to start a component in your own app (for example, to move between screens in the user interface), because you already know the package and class name of that component.
- *Implicit intents* do not specify a specific activity or other component to receive the intent. Instead you declare a general action to perform in the intent. The Android system matches your request to an activity or other component that can handle your requested action. You'll learn more about implicit intents in a later chapter.

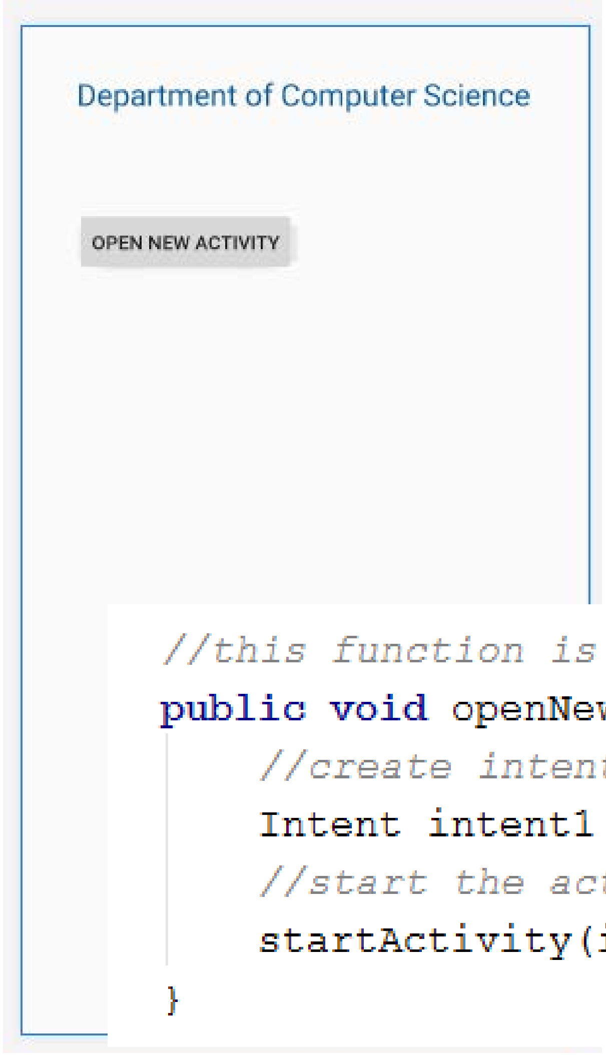
An Example Project

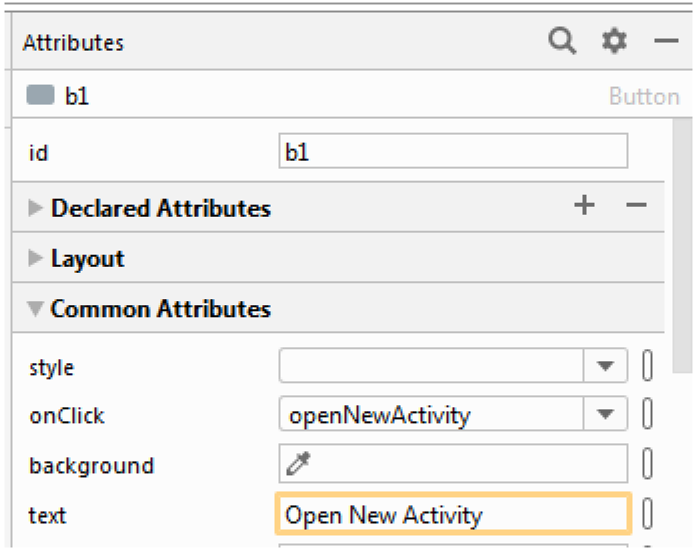
- Create a normal android project.
- In the Main Activity, add Open New Activity button.
- Add a onClick function as “openNewActivity”.
- Add another activity by

  File -> New -> Activity -> Empty Activity
- Name it Activity2

- In the Activity2 layout, add another button with label “Back Button”
- In the onClick attribute add “goBack”
- Create the goBack event handler by Alt+Enter
- Create Intent for the MainActivity.class


activity_main.xml
MainActivity.java
activity_2.xml
Activity2.java
Palette
Common
Text
Buttons
Widgets
TextView
Plain Text
Password
Password (Num...
Pixel
29
AppTheme
Def
0dp




Attributes

b1 Button

id	b1

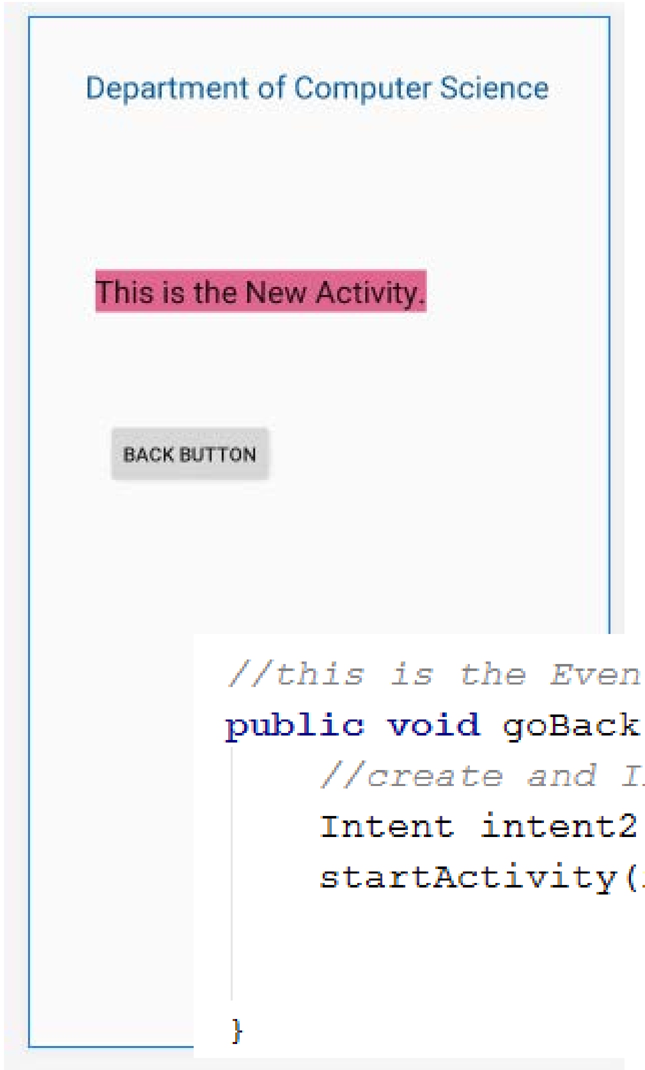
Declared Attributes

Layout

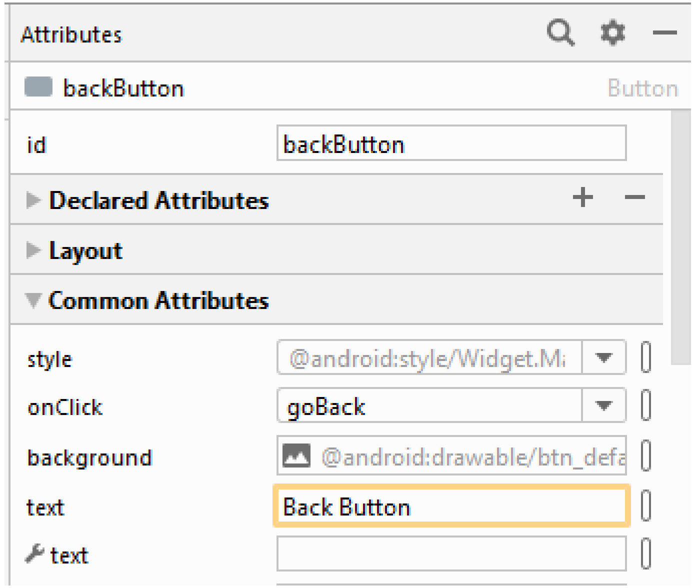
Common Attributes

style	
onClick	openNewActivity
background	
text	Open New Activity

```java
//this function is the Event Handler for the Open New Activity button
public void openNewActivity(View view) {
    //create intent for the Activity2.class
    Intent intent1 = new Intent( packageContext: this, Activity2.class);
    //start the activity intent
    startActivity(intent1);
}
```

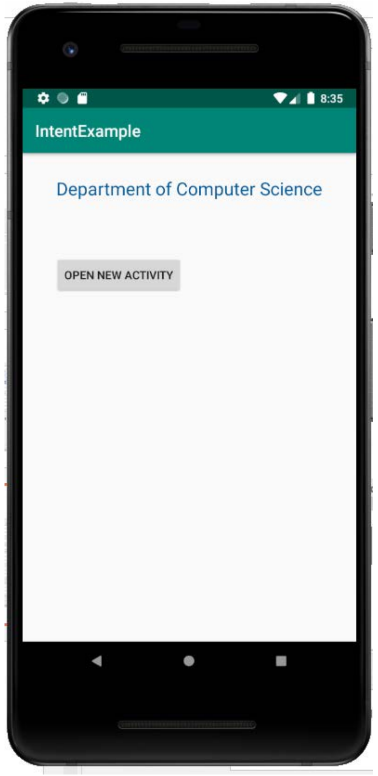

Department of Computer Science

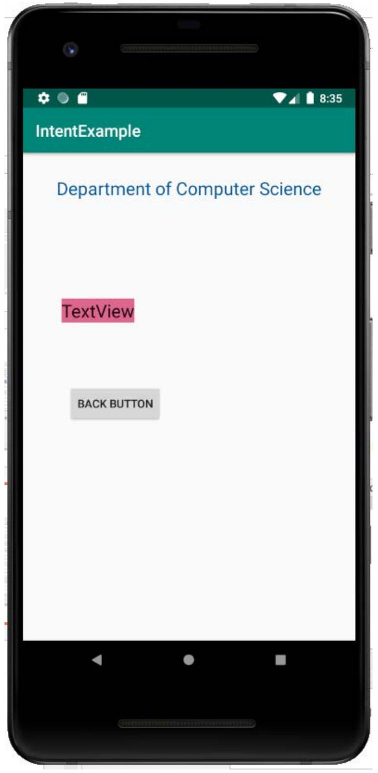
This is the New Activity.



id: backButton
style: @android:style/Widget.M
onClick: goBack
background: @android:drawable/btn_defa
text: Back Button
text:

```java
//this is the Event Handler for the Back Button
public void goBack(View view) {
    //create and Intent for opening the MainActivity.class
    Intent intent2 = new Intent( packageContext: this, MainActivity.class);
    startActivity(intent2);


}
```


8:35
IntentExample
Department of Computer Science
OPEN NEW ACTIVITY


8:35
IntentExample
Department of Computer Science
TextView
BACK BUTTON

Sending Data to other activities through Intent

Two types of sending data with intents

- The Intent Extras
  - These are key-value pairs that carry information the receiving activity requires to accomplish the requested action.
- The Intent data.
  - The intent data field contains a reference to the data you want the receiving activity to operate on, as a Uri object.
  - one piece of information whose data location can be represented by an URI

Use the intent data field:

- When you only have one piece of information you need to send to the started activity.
- When that information is a data location that can be represented by a URI.

Use the intent extras:

- If you want to pass more than one piece of information to the started activity.
- If any of the information you want to pass is not expressible by a URI.

Sending data to another activity

In the first (or sending) `Activity`:

1. Create the `Intent` object
2. Put data or extras into that `Intent`
3. Start the new `Activity` with `startActivity()`

Retrieving data in the Reciever Activity

In the second (or receiving) `Activity`:

1. Get the `Intent` object, the `Activity` was started with by using the getIntent() method.

2. Retrieve the data or extras from the `Intent` object using getData() or getExtras() methods.

Add extras to the intent

To add intent extras to an explicit intent from the originating activity:

1. Determine the keys to use for the information you want to put into the extras, or define your own. Each piece of information needs its own unique key.
2. Use the putExtra() methods to add your key/value pairs to the intent extras. Optionally you can create a Bundle object, add your data to the bundle, and then add the bundle to the intent.

Put information into intent extras

- Passing One value through variable

  putExtra(name, value)

  ```
  intent.putExtra("level", 406);
  intent.putExtra(“dataName", “Value to be passed”);
  ```

- Passing Multiple values through array

  putExtra(String name, String[] value)

  ```
  String[] foodList = {"Rice", "Beans", "Fruit"};
  intent.putExtra("food", foodList);
  ```

- If data is a lot then, then first create a “bundle” of data and then pass the bundle.

Create bundle and fill with data

```
putExtras(bundle);
```

Sending data to an activity with extras

- **In the example below a string is passed to another activity with a name SecondActivity**

```
Intent intent = new Intent(  this,   SecondActivity.class);

String message = "Hello Activity";

intent.putExtra(“data”, message);

startActivity(intent);
```

Example of putting a URI as intent data

```
// A web page URL
intent.setData(Uri.parse(http://www.google.com) );

// a Sample file URI
intent.setData( Uri.fromFile(
          new File("/sdcard/sample.jpg")));
```

Get data from intents

- Get Bundle: `getExtras()` Get all the data at once as a bundle.

```
Bundle bundle = intent.getExtras();
```

- Retreiving a URI: `getData();`

```
Uri locationUri = intent.getData();
```

There some are other function as as well.

Example.

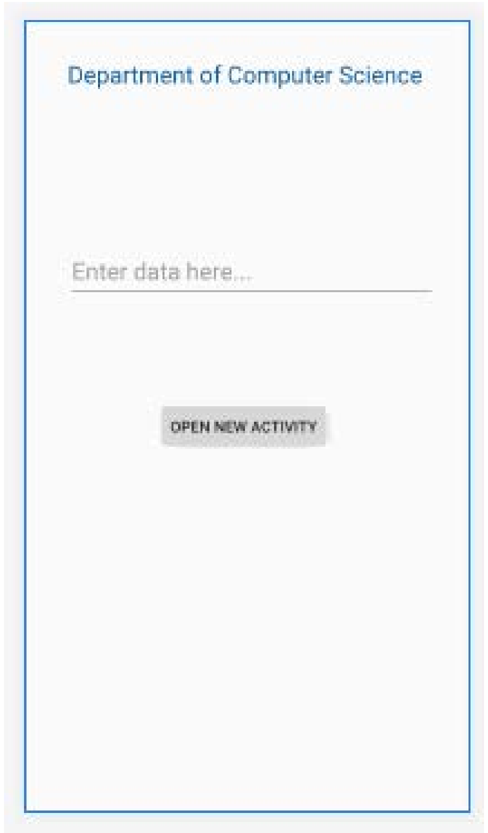
In the Calling Activity

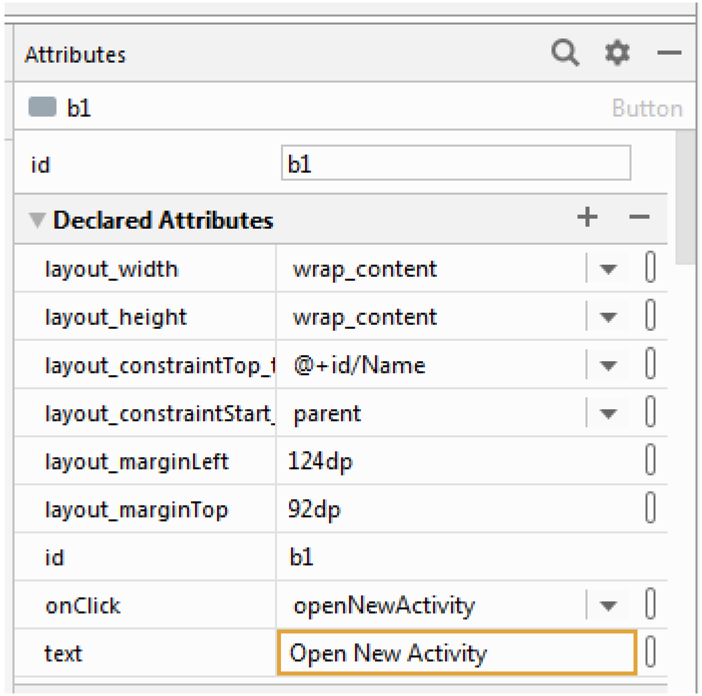
```
public void callSecondActivity(View view){
    Intent i = new Intent(getApplicationContext(), SecondActivity.class);
    i.putExtra("Value1", "Android By Javatpoint");
    i.putExtra("Value2", "Simple Tutorial");
    // Set the request code to any code you like, you can identify the
    // callback via this code
    startActivity(i);
}
```

In the Called Activity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    Bundle extras = getIntent().getExtras();
    String value1 = extras.getString("Value1");
    String value2 = extras.getString("Value2");
    Toast.makeText(getApplicationContext(),"Values are:\n First value: "+value1+
            "\n Second Value: "+value2, Toast.LENGTH_LONG).show();
}
```

MainActivity





Attributes

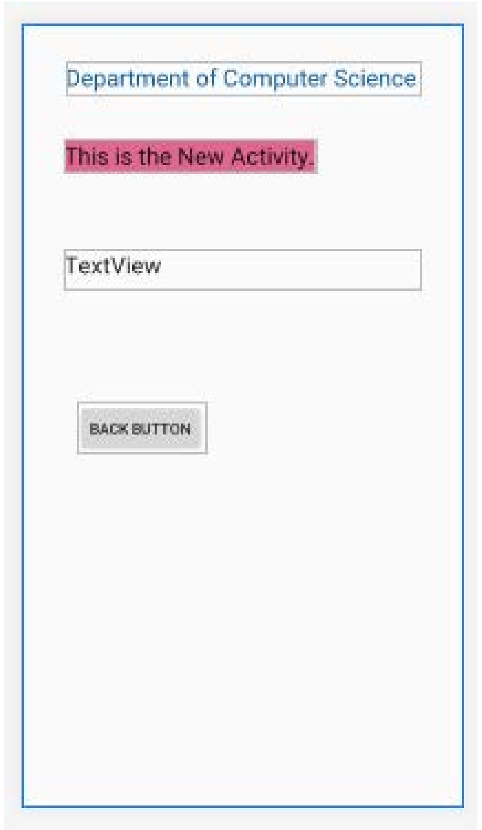
b1 Button

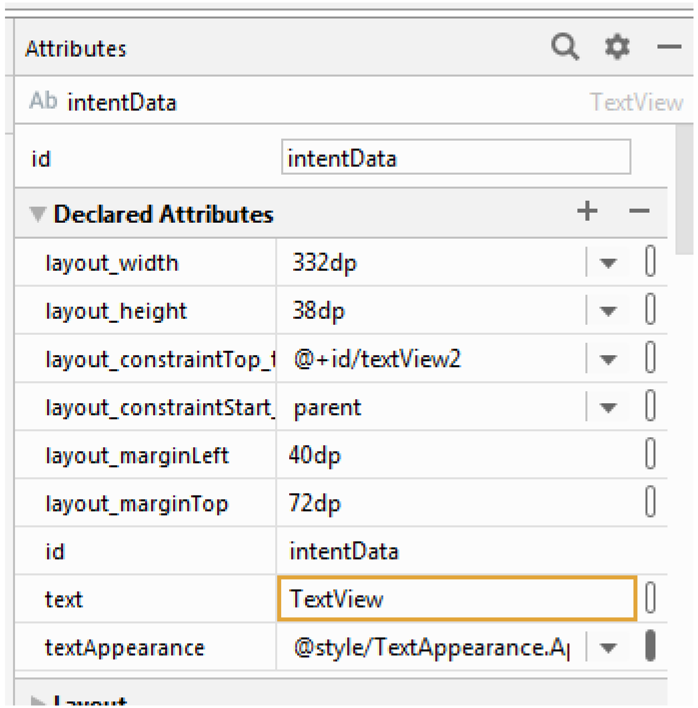
id	b1

Declared Attributes

layout_width	wrap_content
layout_height	wrap_content
layout_constraintTop_t	@+id/Name
layout_constraintStart_	parent
layout_marginLeft	124dp
layout_marginTop	92dp
id	b1
onClick	openNewActivity
text	Open New Activity

Activity2




Attributes

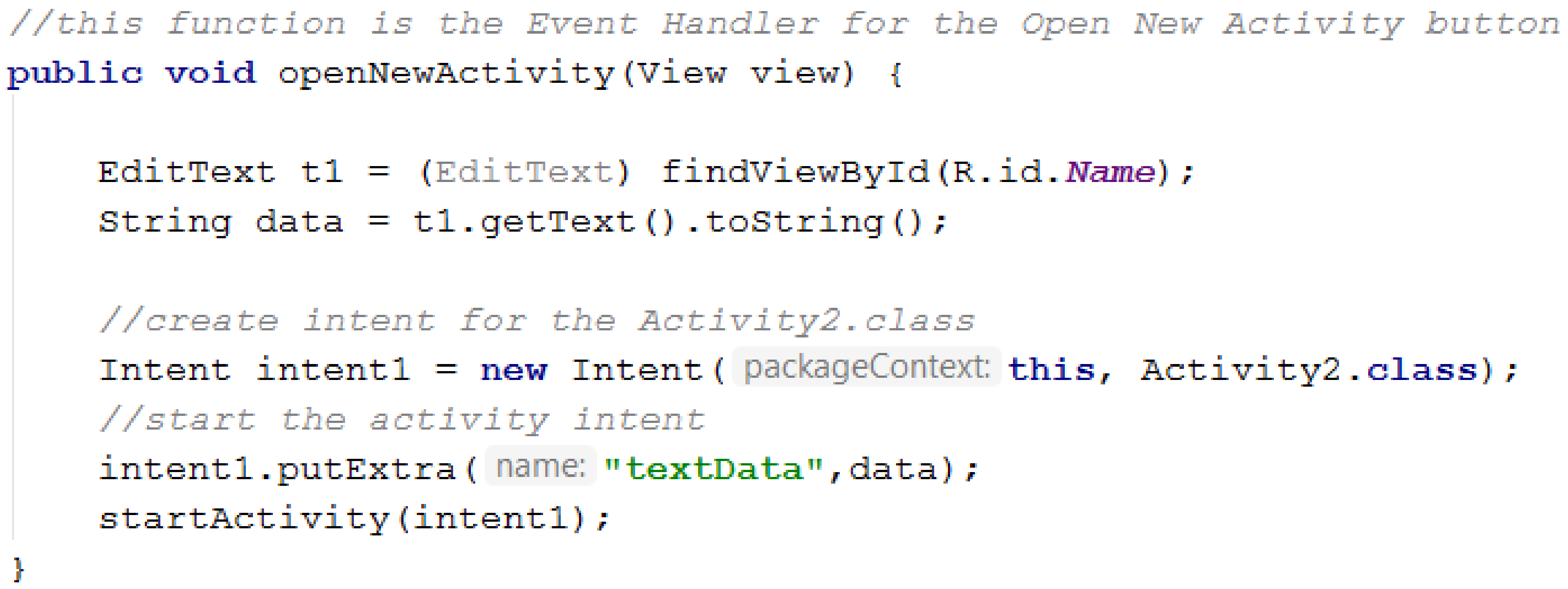
Ab intentData TextView

id intentData

Declared Attributes

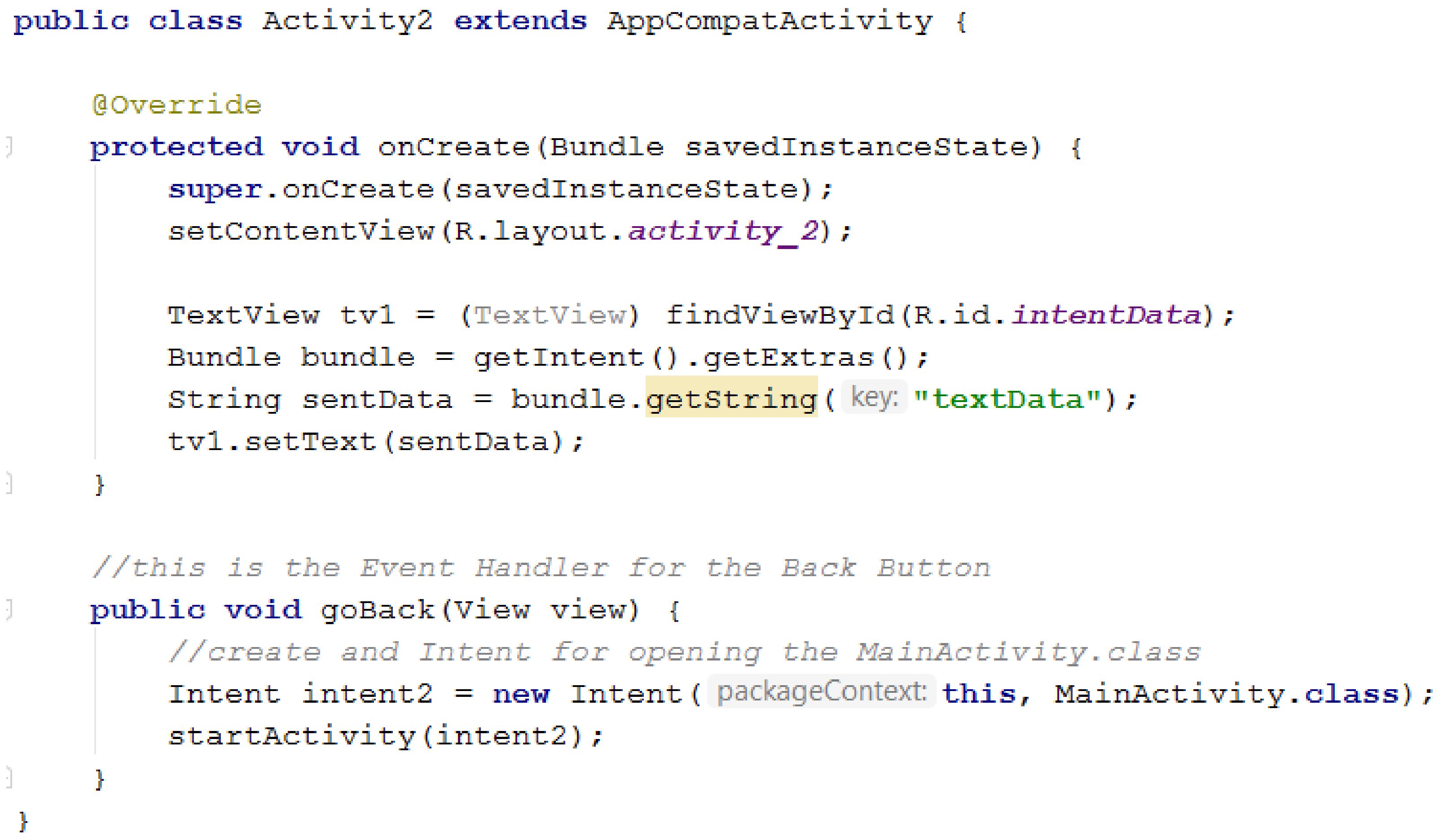
layout_width	332dp
layout_height	38dp
layout_constraintTop_t	@+id/textView2
layout_constraintStart_	parent
layout_marginLeft	40dp
layout_marginTop	72dp
id	intentData
text	TextView
textAppearance	@style/TextAppearance.A

MainActivity: Button Event Handler

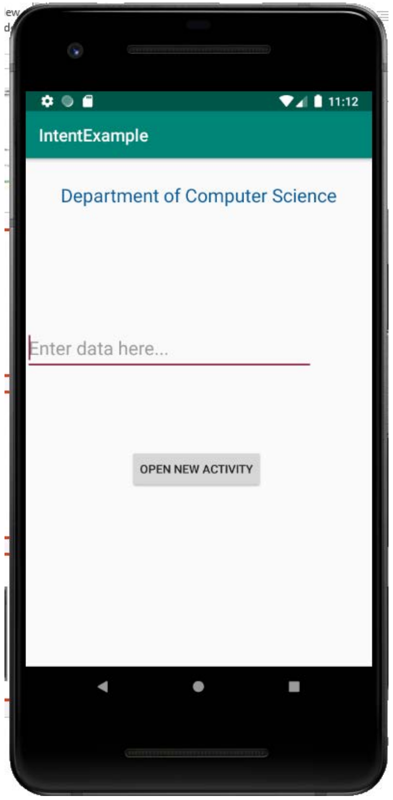


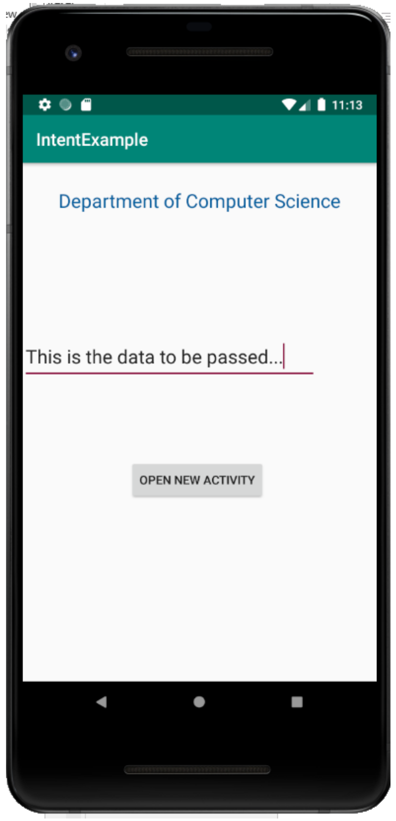
```java
//this function is the Event Handler for the Open New Activity button
public void openNewActivity(View view) {

    EditText t1 = (EditText) findViewById(R.id.Name);
    String data = t1.getText().toString();

    //create intent for the Activity2.class
    Intent intent1 = new Intent( packageContext: this, Activity2.class);
    //start the activity intent
    intent1.putExtra( name: "textData",data);
    startActivity(intent1);
}
```

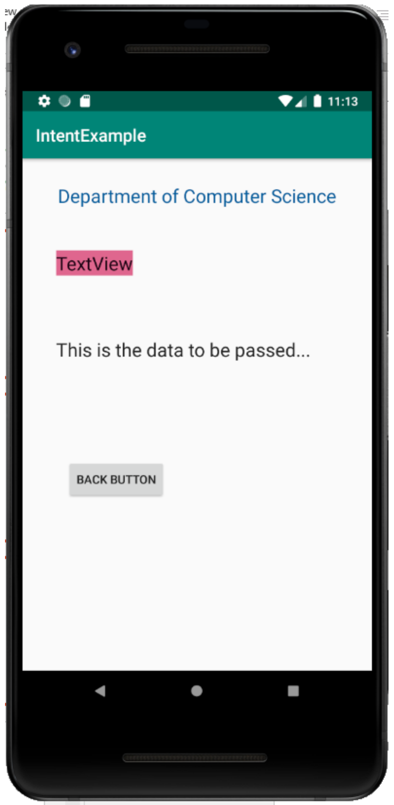
Activity2.java



```
public class Activity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_2);

        TextView tv1 = (TextView) findViewById(R.id.intentData);
        Bundle bundle = getIntent().getExtras();
        String sentData = bundle.getString( key: "textData");
        tv1.setText(sentData);
    }

    //this is the Event Handler for the Back Button
    public void goBack(View view) {
        //create and Intent for opening the MainActivity.class
        Intent intent2 = new Intent( packageContext: this, MainActivity.class);
        startActivity(intent2);
    }
}
```


11:12
IntentExample
Department of Computer Science
Enter data here...
OPEN NEW ACTIVITY


11:13
IntentExample
Department of Computer Science
This is the data to be passed...
OPEN NEW ACTIVITY


11:13
IntentExample
Department of Computer Science
TextView
This is the data to be passed...
BACK BUTTON