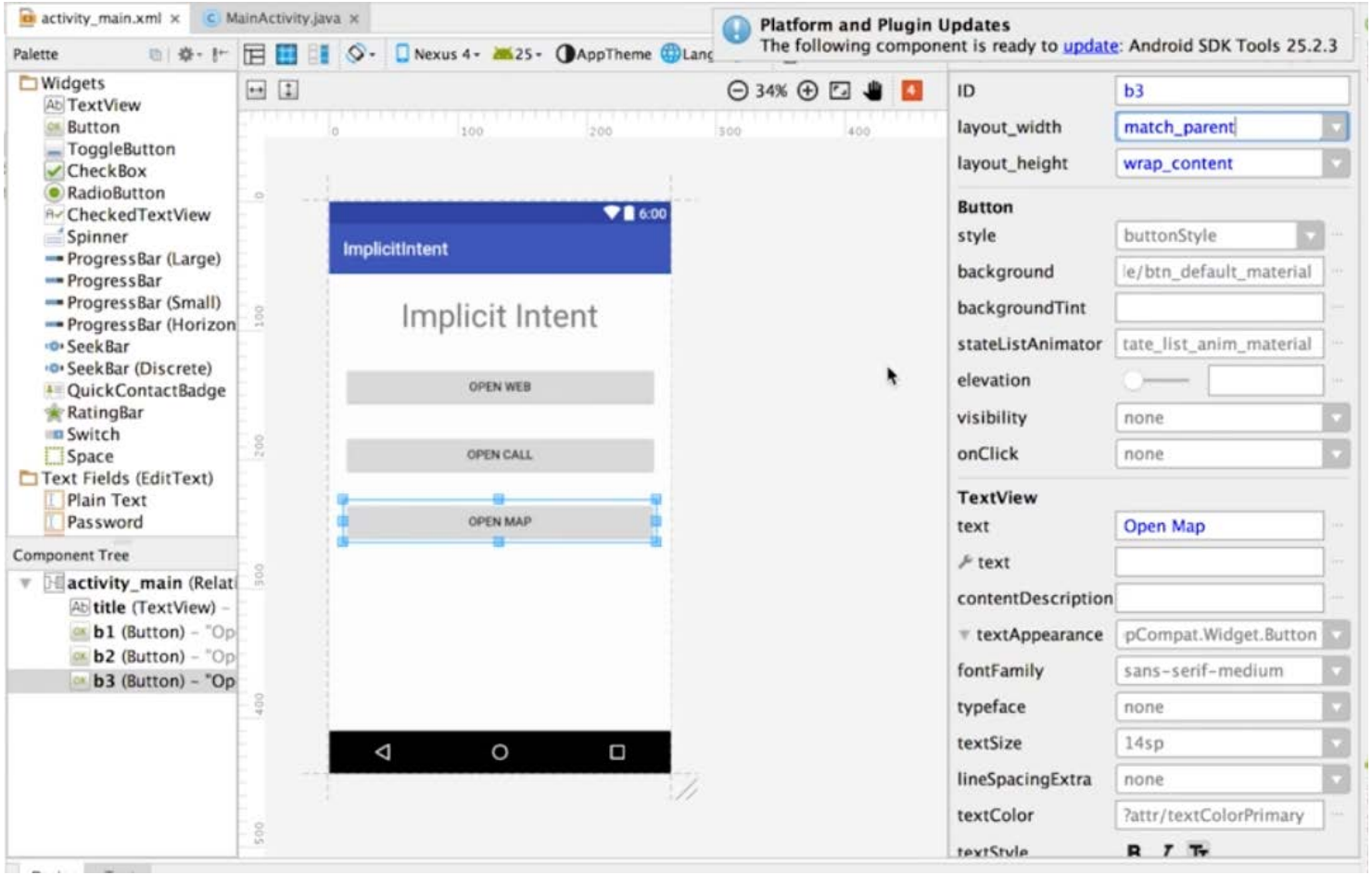
Mobile Application Development

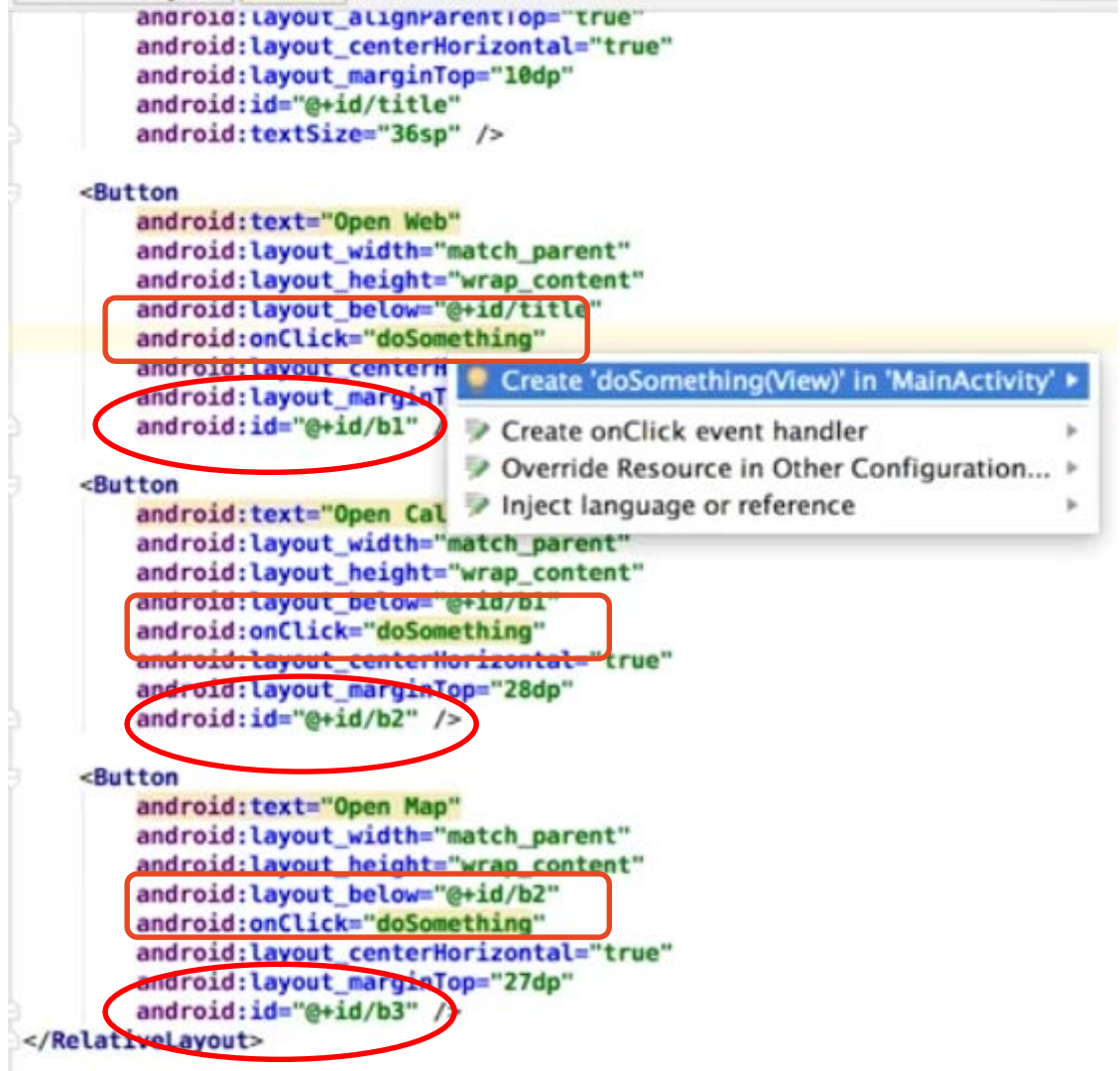
Lesson 7

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

- Implicit Intent Example
- Returning Data from Activities

Implicit Intent Example





The three buttons have ids as

- b1
- b2
- B3

All the three buttons have the same onClick event handler function name “doSomething”

```java
package com.example.demouser.implicitintent;

import ...

public class MainActivity extends AppCompatActivity {

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

    public void doSomething(View view) {

        switch (view.getId()){

            case R.id.b1 :

                break;

            case R.id.b2 :

                break;

            case R.id.b3:

                break;

        }
    }
}
```

In the doSomething() function, first check the ID of the button that called the function.

Then do the respective task.

xample > demouser > implicitintent > MainActivity

tivity_main.xml × | MainActivity.java ×

Platform and Plugin Up
The following componen

```
package com.example.demouser.implicitintent;

import ...

public class MainActivity extends AppCompatActivity {

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

    public void doSomething(View view) {

        switch (view.getId()){

            case R.id.b1 :
                Intent i1 = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com"));
                startActivity(i1);
                break;

            case R.id.b2 :

                break;

            case R.id.b3:

                break;

        }
    }
}
```

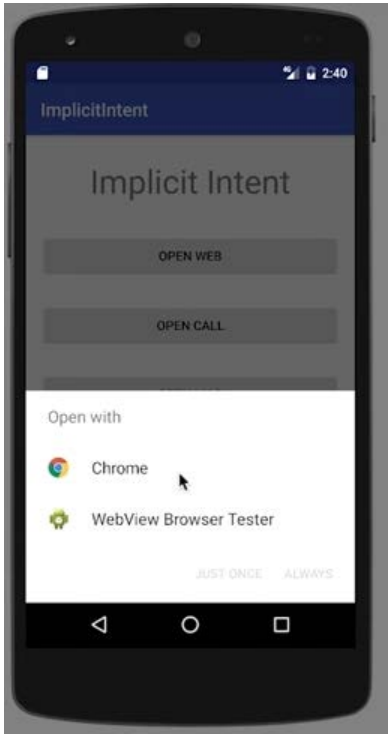
ctivity_main.xml × MainActivity.java ×

Platform and Plugin
The following compor

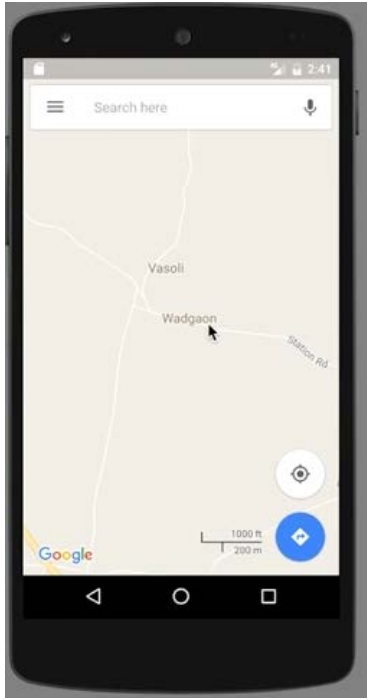
```
package com.example.demouser.implicitintent;

import ...

public class MainActivity extends AppCompatActivity {

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

    public void doSomething(View view) {

        switch (view.getId()){

            case R.id.b1 :
                Intent i1 = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com"));
                startActivity(i1);
                break;

            case R.id.b2 :
                Intent i2 = new Intent(Intent.ACTION_VIEW,Uri.parse("tel:971297873423"));
                startActivity(i2);
                break;

            case R.id.b3:
                Intent i3 = new Intent(Intent.ACTION_VIEW,Uri.parse("geo:20.5937,78.9629"));
                startActivity(i3);
                break;

        }
    }
}
```


2:39
ImplicitIntent
Implicit Intent
OPEN WEB
OPEN CALL
OPEN MAP

OPEN WEB button clicked



Open Call and Open MAP Buttons Clicked



Returning Data from an Activity

1. So far we have seen that we have passed data to an activity
2. But in many cases we also get back data from another activity. For example, choosing a file to attach, an activity takes a picture and returns the picture. etc.

General STEPS

1. Use `startActivityForResult()` function instead of StartActivity().
2. The target activity returns data through Intent
3. Receive the data in the source activity in the function `onActivityResult()`.

Sometimes when you send data to an activity with an intent, you would like to also get data back from that intent. For example, you might start a photo gallery activity that lets the user pick a photo. In this case your original activity needs to receive information about the photo the user chose back from the launched activity.

To launch a new activity and get a result back, do the following steps in your originating activity:

1. Instead of launching the activity with startActivity(), call startActivityForResult() with the intent and a request code.
2. Create a new intent in the launched activity and add the return data to that intent.
3. Implement onActivityResult() in the originating activity to process the returned data.

Returning data to the starting activity

1. First, Use `startActivityForResult()` to start the second `Activity`
2. To return data from the second `Activity`:
   - Create a ***new*** **`Intent`**
   - Put the response data in the **`Intent`** using **`putExtra`**`()`
   - Set the result to **`Activity.RESULT_OK`**
     or **`RESULT_CANCELED`**, if the user cancelled out
   - **call `finish()`** to close the `Activity`
3. Then Implement **`onActivityResult()`** in first **`Activity`**

Working of startActivityForResult()

`startActivityForResult(intent, requestCode);`

- It starts an `Activity` through `intent`. It assigns it an integer identifier (`requestCode`). This would identify the request.
- Returns data via `Intent` extras
- When done, pop stack, return to previous `Activity`, and execute `onActivityResult()` callback to process returned data, and using `requestCode` to identify which `Activity` has "returned“.

To get data back from a launched activity, start that activity with the startActivityForResult() method instead of startActivity().

```
startActivityForResult(messageIntent, TEXT_REQUEST);
```

The startActivityForResult() method, like startActivity(), takes an intent argument that contains information about the activity to be launched and any data to send to that activity. The startActivityForResult() method, however, also needs a request code.

The request code is an integer that identifies the request and can be used to differentiate between results when you process the return data. For example, if you launch one activity to take a photo and another to pick a photo from a gallery, you'll need different request codes to identify which request the returned data belongs to.

Conventionally you define request codes as static integer variables with names that include REQUEST. Use a different integer for each code. For example:

```
public static final int PHOTO_REQUEST = 1;
public static final int PHOTO_PICK_REQUEST = 2;
public static final int TEXT_REQUEST = 3;
```

Example: STEP1: startActivityForResult()

```
public static final int TEXT_REQUEST = 1;
Intent intent = new Intent(this, ChooseFoodItemsActivity.class);
startActivityForResult(intent, TEXT_REQUEST);
```

In the above example, TEXT_REQUEST constant is used for identifier. The request code 1 is assigned to the intent that will send back some data.

STEP2: Return data and finish second activity

The following code goes in to the Second Activity.

```
// Create an intent
Intent replyIntent = new Intent();
// Put the data to return into the extra
replyIntent.putExtra(REPLY_KEY, reply_data_value);
// Set the activity's result to RESULT_OK
setResult(RESULT_OK, replyIntent);
// Finish the current activity
finish();
```

STEP3: Implement onActivityResult()

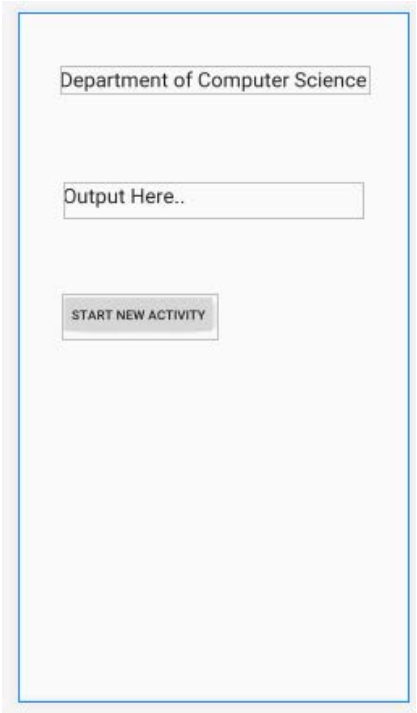
```
public void onActivityResult(int requestCode,
                             int resultCode, Intent data) {
  super.onActivityResult(requestCode, resultCode, data);
  if (requestCode == TEXT_REQUEST) { // Identify activity
    if (resultCode == RESULT_OK) { // Activity succeeded
      String reply = data.getStringExtra(SecondActivity.REPLY_KEY);
      // … use the data here
 }}}
```

The three arguments to the onActivityResult() contain all the information you need to handle the return data.

- **Request code.** The request code you set when you launched the activity with startActivityForResult(). If you launch different activities to accomplish different operations, use this code to identify the specific data you're getting back.
- **Result code:** the result code set in the launched activity, usually one of RESULT_OK or RESULT_CANCELED.
- **Intent data**. the intent that contains the data returned from the launch activity.

The example method shown above shows the typical logic for handling the request and response codes. The first test is for the TEXT_REQUEST request, and that the result was successful. Inside the body of those tests you extract the return information out of the intent. Use getData() to get the intent data, or getExtra() to retrieve values out of the intent extras with a specific key.

Calling Activity

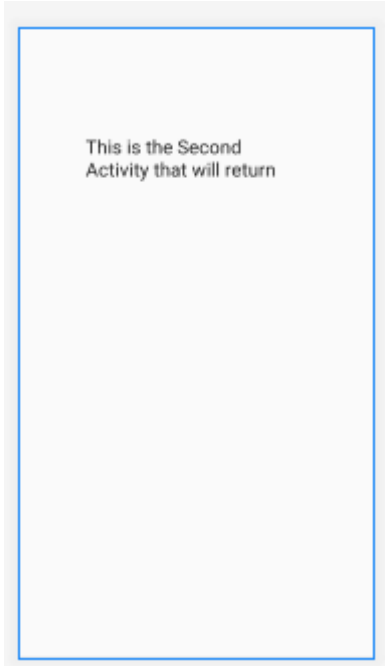



```
<TextView
    android:id="@+id/output"
    android:layout_width="321dp"
    android:layout_height="38dp"
    android:layout_marginTop="95dp"
    android:text="Output Here.."
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    tools:layout_editor_absoluteX="48dp" />

<Button
    android:id="@+id/b1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="81dp"
    android:onClick="openActivity"
    android:text="Start New Activity"
    app:layout_constraintTop_toBottomOf="@+id/output"
    tools:layout_editor_absoluteX="46dp" />
```

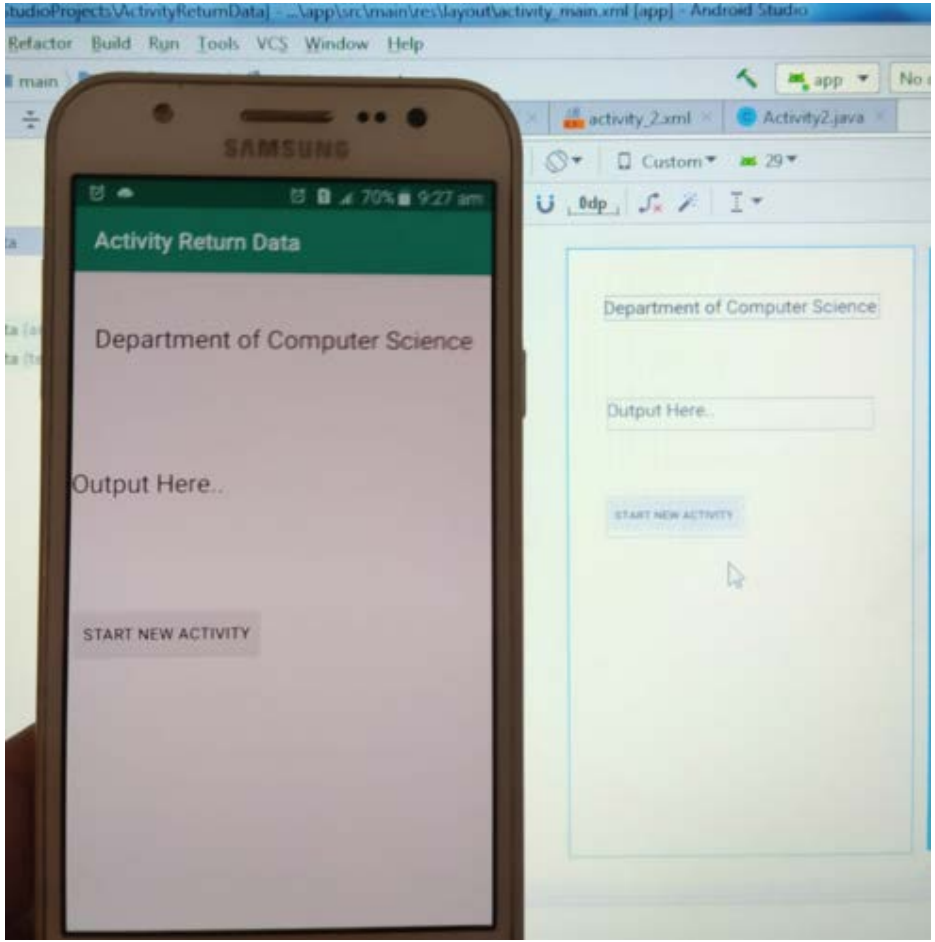
Second Activity

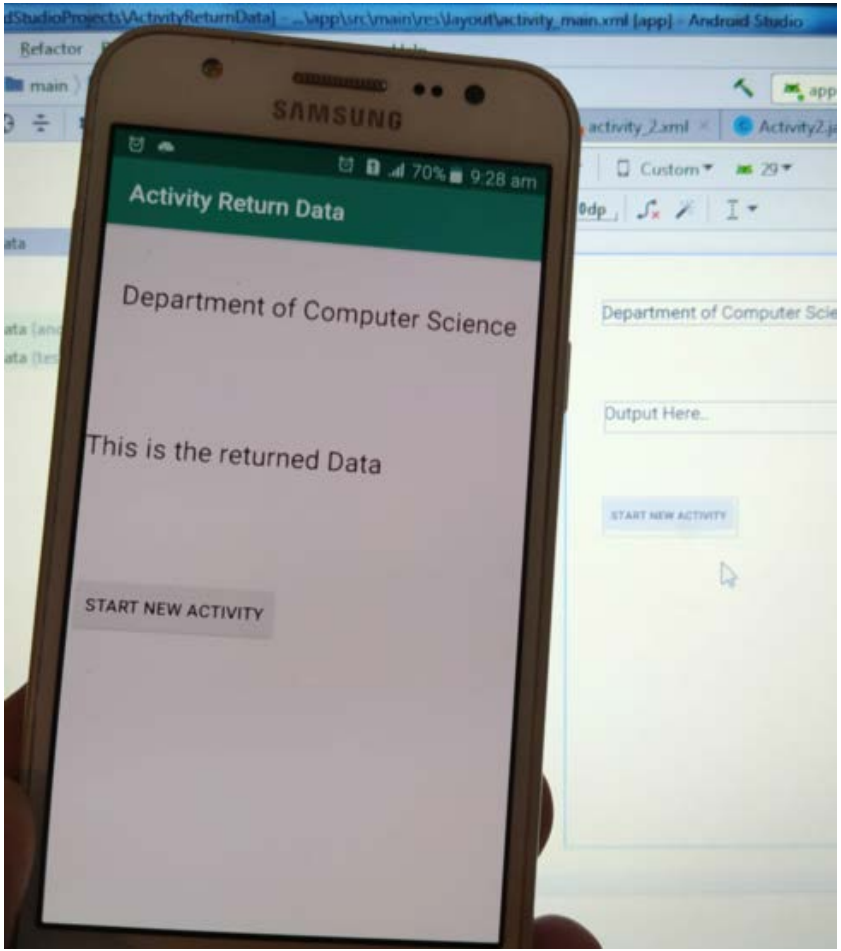
```
public static final int REQUEST_CODE = 1;

public void openActivity(View view) {
    Intent i = new Intent( packageContext: this,Activity2.class);
    startActivityForResult(i,REQUEST_CODE);
}

public void onActivityResult(int reqCode, int resultCode, Intent data)
{
    super.onActivityResult(reqCode,resultCode,data);
    if (reqCode == REQUEST_CODE)
    {
        if (resultCode == RESULT_OK)
        {
            String result = data.getStringExtra( name: "retData");
            TextView output = findViewById(R.id.output);
            output.setText(result);

        }
    }
}
```

Calling Activity




```
public class Activity2 extends AppCompatActivity {

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

        Intent i2 = new Intent();
        i2.putExtra( name: "retData", value: "This is the returned Data");
        setResult(RESULT_OK, i2);
        finish();
    }
}
```


Activity Return Data
Department of Computer Science
Output Here..
START NEW ACTIVITY


Activity Return Data
Department of Computer Science
This is the returned Data
START NEW ACTIVITY