Lecture 3
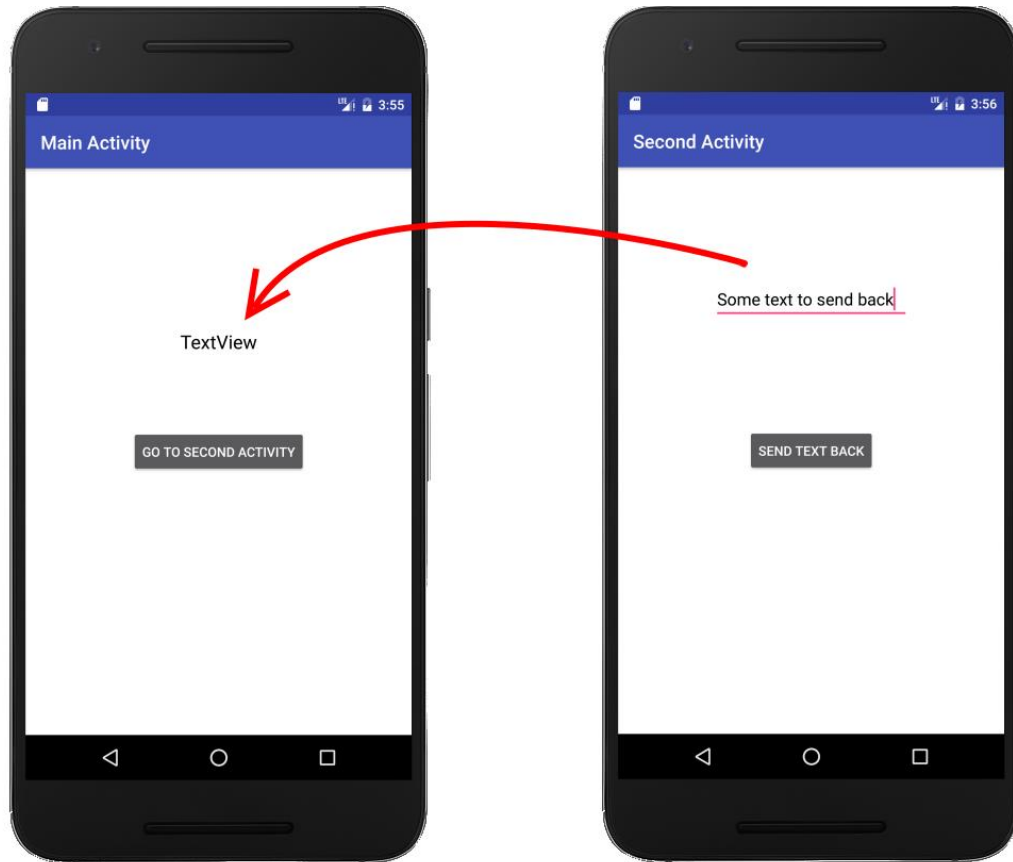# Intents

Course: **Mobile App Development**
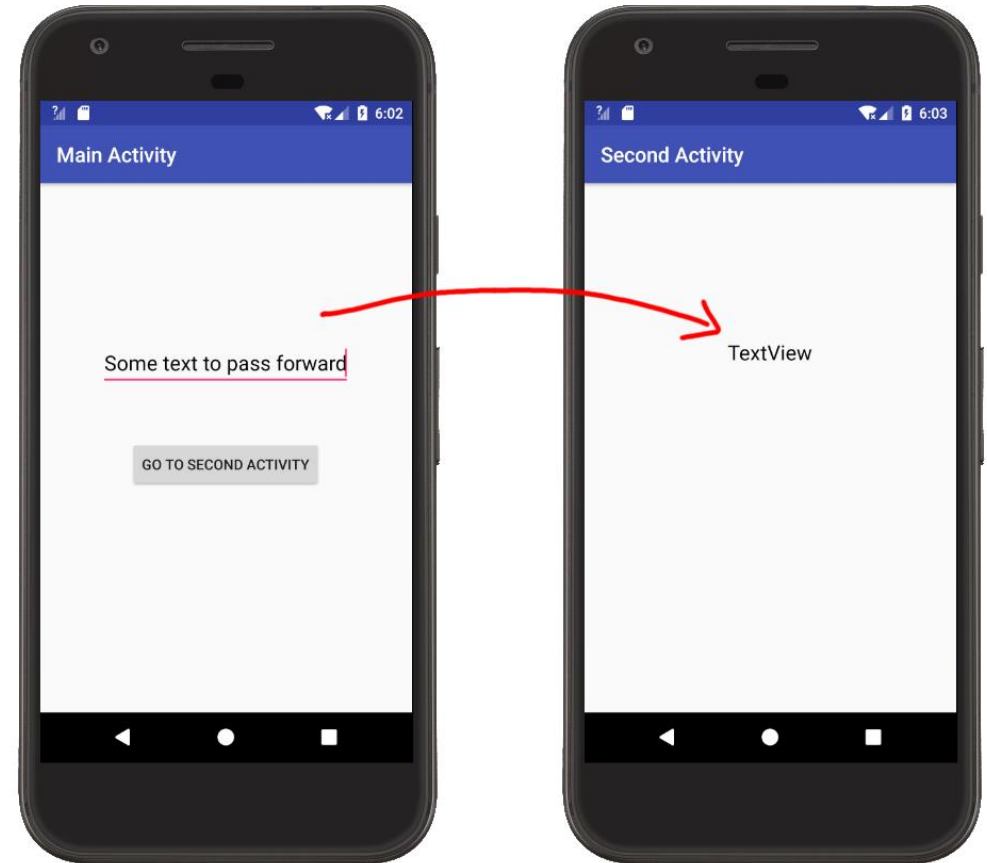
By: Tho C. Mai
Nha Trang University

Intents ..

- Returning data from an activity

☐ Passing data to an activity

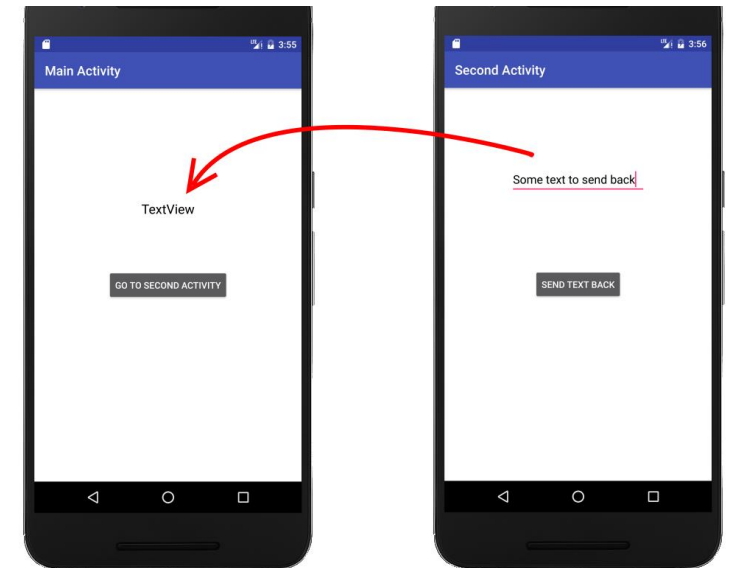# Returning data from an activity

First Activity

startActivityForResult()

Second Activity

- Create an Intent object   (ex: iData)
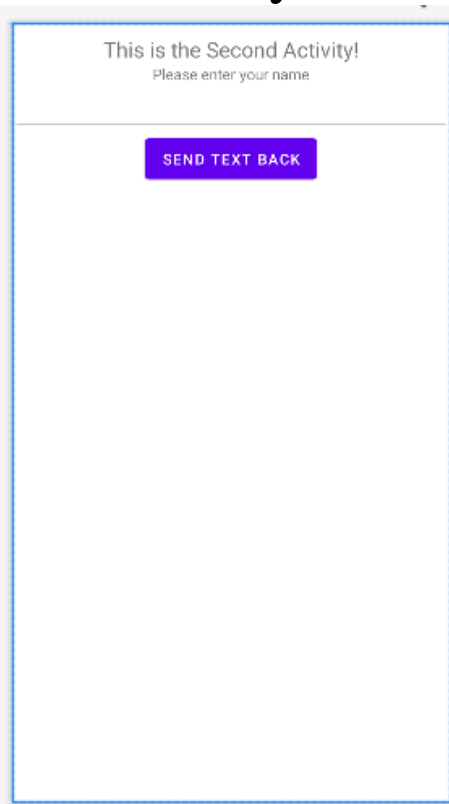
- iData.putExtra("key", value)

onActivityResult()

- Get value: data.getDataTypeExtra("key")

Returning data from an activity

- Modify the <u>UsingIntent</u> app
- First, modify the <u>activity_second</u>.xml:

This is the Second Activity!
Please enter your name

SEND TEXT BACK

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://sche
    xmlns:app="http://schemas.android.com
    xmlns:tools="http://schemas.android.co
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".SecondActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content
        android:paddingTop="10dp"
        android:textSize="20sp"
        android:gravity="center"
        android:text="This is the Second Activi
```

```xml
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Please enter your name"
    android:id="@+id/textView3" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/edtUsername" />
<Button
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:text="Send text back"
    android:layout_gravity="center"
    android:id="@+id/btnSentDataBack"
    android:onClick="onClick"/>
</LinearLayout>
```

Returning data from an activity

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

```java
….
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {  ..  }
    public void onClick(View view) {
        Intent intentBacktoMainActivity = new Intent();
        //---get the EditText view---
        EditText txt_username = (EditText)findViewById(R.id.edtUsername);
        //---put/set the data to pass back, in key,value pair format---
        intentBacktoMainActivity.putExtra("UNAME" , txt_username.getText().toString());
        setResult(RESULT_OK, intentBacktoMainActivity);
        //---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 putExtra() 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:

..

```java
int request_Code = 1;
@Override
protected void onCreate(Bundle savedInstanceState) { ...}
public void onClick(View view) {
    Intent intent_for_SecondActiviy = new Intent(this, SecondActivity.class);
    startActivityForResult(intent_for_SecondActiviy, request_Code);
}
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == request_Code) {
        if (resultCode == RESULT_OK) {
            Toast.makeText(this, data.getStringExtra("UNAME").toString(), Toast.LENGTH_LONG).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

To retrieve the data set using the putExtra () method, use the getDatatypeExtra() method
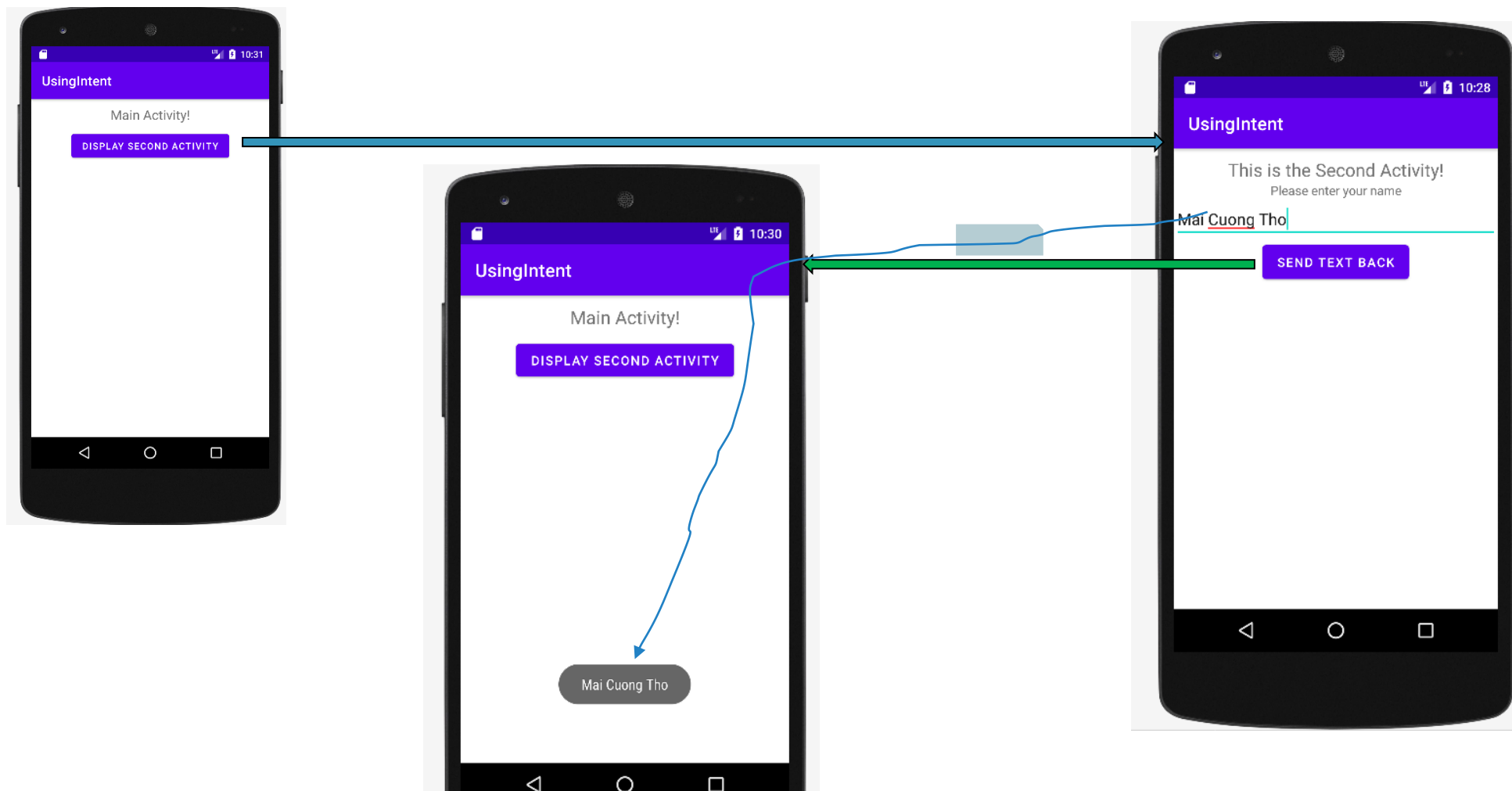
Toast

- 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_LONG Show the view or text notification for a long period of time.
int	LENGTH_SHORT Show the view or text notification for a short period of time.

Returning data from an activity

Passing data to an activity

First Activity

String
Int
Intent.putExtra()   Bundle

startActivityForResult()
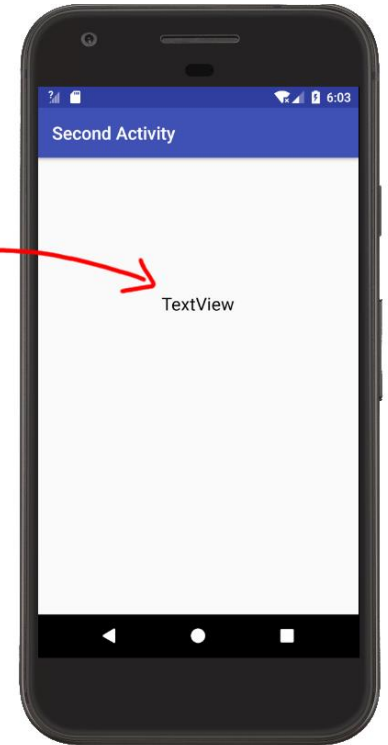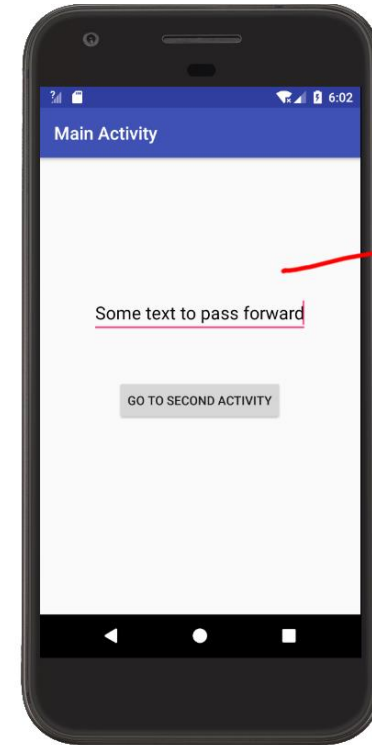
Second
Activity

Retrieve data   **Pass data back**
getIntent().getStringExtra()   Intent.putExtra()
getIntent().getIntExtra()
getIntent().getExtra()
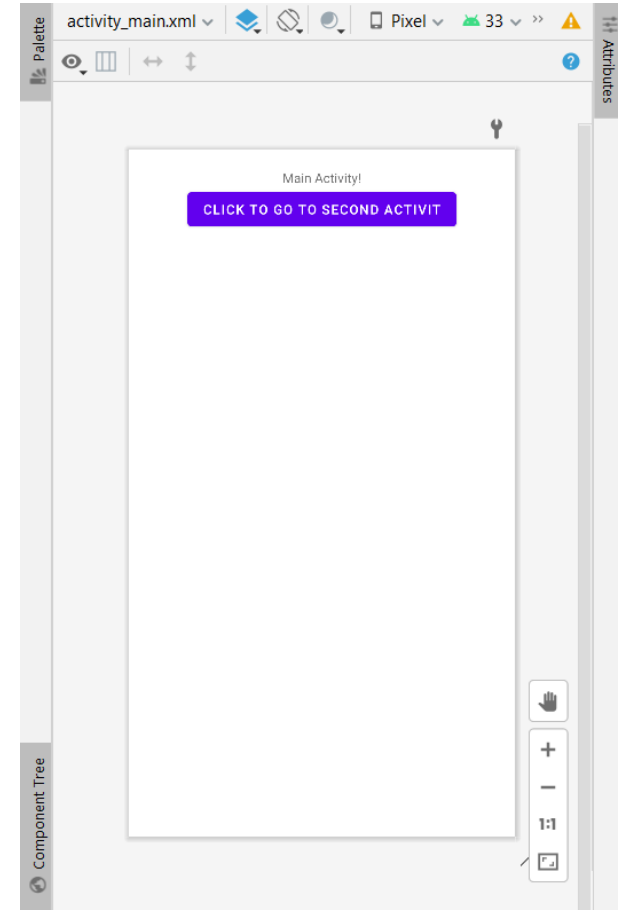[for Bundle]   Intent.setData()

onActivityResult()

Retrieve Data
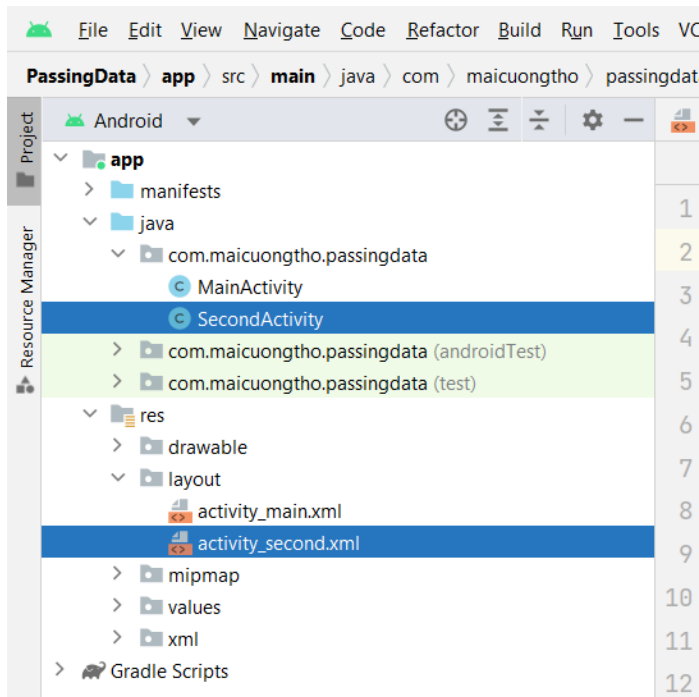
Passing data to an activity

- Create a new project and name it **PassingData**

- Add/replace with the bolded statements in the following code to the _activity_main.xml_ file:

```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:layout_gravity="center"
        android:paddingTop="20dp"
        android:text="Main Activity!" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click to go to Second Activit"
        android:onClick="onClick"
        android:layout_gravity="center"
        />
</LinearLayout>
```
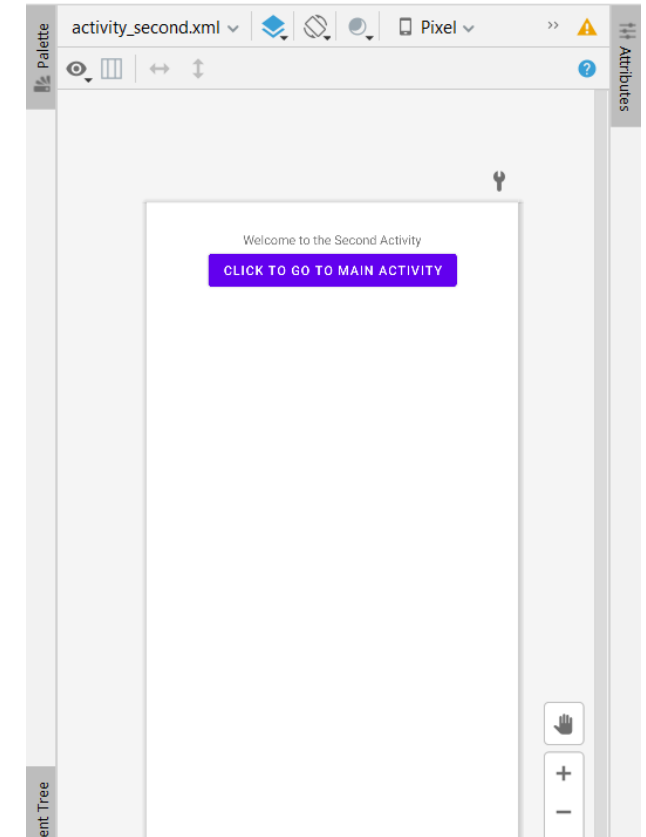
Passing data to an activity

- Add a new SecondActivity, and modify *activity_second.xml* as follows:


```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=".SecondActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to the Second Activity"
        android:id="@+id/textView"
        android:paddingTop="30dp"
        android:layout_gravity="center" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click to go to Main Activity"
        android:onClick="onClick"
        android:layout_gravity="center"/>
</LinearLayout>
```

Passing data to an activity

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

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.

```java
..
public class MainActivity extends AppCompatActivity {
    ....
    public void onClick(View view) {
        Intent i = new Intent(this, SecondActivity.class);
//---use putExtra() to add new name/value pairs---
        i.putExtra("name1", "Mai Cuong Tho");
        i.putExtra("age1", 25);
//---use a Bundle object to add new name/values pairs---
        Bundle extras = new Bundle();
        extras.putString("name2", "Pham Hong Tuoi");
        extras.putInt("age2", 35);
//---attach the Bundle object to the Intent object---
        i.putExtras(extras);
//---start the activity ( don't get a result back)---
        startActivity(i);
    }

}
```

Passing data to an activity

- Populate the **SecondActivity.java** file as follows:

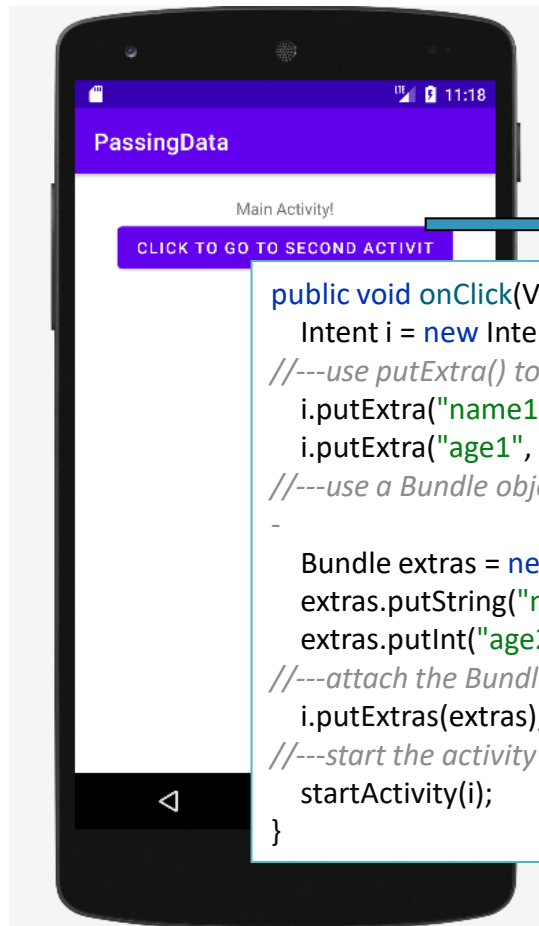> To obtain the data sent using the Intent object,
> - you first obtain the Intent object using the getIntent() method.
> - Then, call its getStringExtra() method to get the string value set using The putExtra() method

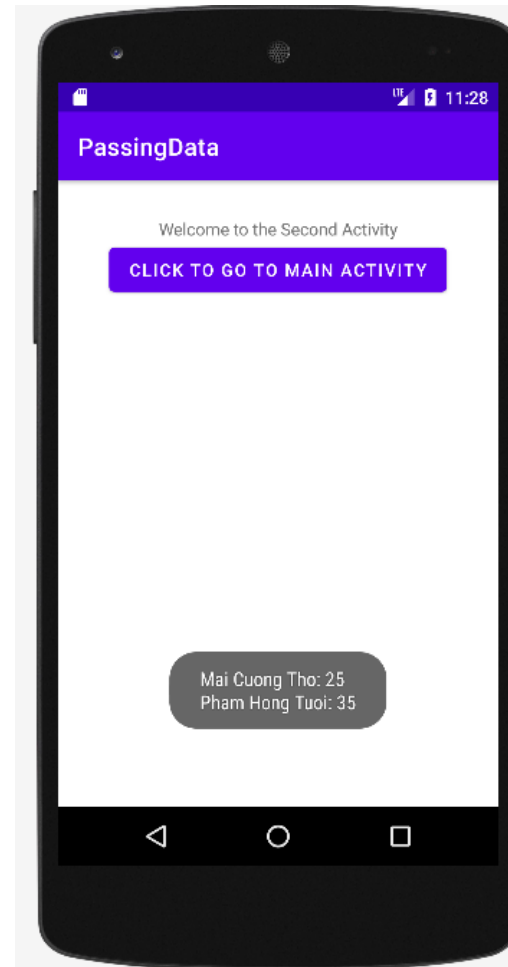> use getInt() method to retrieve an integer value


```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    // --- get the intent sent from MainActivity
    Intent intentReceived = getIntent();
    //---get the data passed in using getStringExtra()---
    String name1= intentReceived.getStringExtra("name1");
    //---get the data passed in using getIntExtra()---
    int age1 = intentReceived.getIntExtra("age1", 0);
    //---get the Bundle object passed in---
    Bundle bundle = intentReceived.getExtras();
    //---get the data using the getString()---
    String name2 = bundle.getString("name2");
    //---get the data using the getInt() method---
    int age2 = bundle.getInt("age2");
    String messageToast = name1 + ": " +String.valueOf(age1) + "\n" +
                name2 + ": " +String.valueOf(age2);
    Toast.makeText(this,messageToast,    Toast.LENGTH_SHORT).show();
}
```

Passing data to an activity


```java
public void onClick(View view) {
    Intent i = new Intent(this, SecondActivity.class);
//---use putExtra() to add new name/value pairs---
    i.putExtra("name1", "Mai Cuong Tho");
    i.putExtra("age1", 25);
//---use a Bundle object to add new name/values pairs---
    Bundle extras = new Bundle();
    extras.putString("name2", "Pham Hong Tuoi");
    extras.putInt("age2", 35);
//---attach the Bundle object to the Intent object---
    i.putExtras(extras);
//---start the activity ( don't get a result back)---
    startActivity(i);
}
```

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    // --- get the intent sent from MainActivity
    Intent intentReceived = getIntent();
    //---get the data passed in using getStringExtra()---
    String name1=
intentReceived.getStringExtra("name1");
    //---get the data passed in using getIntExtra()---
    int age1 = intentReceived.getIntExtra("age1", 0);
    //---get the Bundle object passed in---
    Bundle bundle = intentReceived.getExtras();
    //---get the data using the getString()---
    String name2 = bundle.getString("name2");
    //---get the data using the getInt() method---
    int age2 = bundle.getInt("age2");
    String messageToast = name1 + ": "
+String.valueOf(age1) + "\n" +
            name2 + ": " +String.valueOf(age2);
    Toast.makeText(this,messageToast,
        Toast.LENGTH_SHORT).show();
}
```

Your turn

I fear not the man who has practiced 10,000 kicks once, but I fear the man who has practiced one kick 10,000 times.

Bruce Lee

- Practice 4/ Excercie 4
    1) Repeat the example by yourself
    2) Push it to your github repository
        ✓ With a report with screenshots of the final app in action, data structures used/class design, and the implementation logic.

- Practice 5/ Excercie 6 (Homework #2)
- Practice 6/ Excercie 6 (Homework #3)

Homework #2

- Develop a contact management app version 1:
  - 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 an "Add Contact" 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 second activity will be closed, and the app goes back to the main activity
  - The main activity now should display the new contact information

What to submit:
  - Push it to your github repository,
    - a report with screenshots of the final app in action, data structures used/class design, and the implementation logic.

Homework #3