**PRACTICAL # 04**

**OBJECT:**

Working with Activities

**THEORY:**

Activities are fundamental to many Android applications. An application typically has one or more activities. Main purpose of an activity is to interact with the user. From the moment an activity appears on the screen to the moment it is hidden, it goes through a number of stages, known as an activity’s life cycle.

When the user switches between different UI screens corresponding to different activities, one activity is stopped, and the other is started.

An activity represent a screen that extends class android.app.Activity or AppCompactActivity

Each activity has an XML layout eg. activity\_main.xml

**Starting Activity**

You can also start activities that belong to other applications, a mail program for instance.

Here we’ll work with activities within our own app.

To start a new activity, call **startActivity()** passing an Intent as the argument:

- The Intent describes the activity, you want to start.

- The user can then select from the possible intents.

- Or the activity just runs an intent e.g. it opens a web browser if intent is to view a webpage.

- An activity can also receive data and return results (Covered in Lab 5).

**Creating a Second Activity**

An activity is just a class, so we just

- add a new activity class

- Inherit from the Activity or AppCompatActivity base class

- Add the activity to the manifest file

Alternatively, we can add an activity through IDE, which will automate these tasks

To add new activity from IDE, choose:

File→ New → Activity → [Activity Type]

[Activity Type]: The type of activity, for example login activity – different types of activities are available to quick start an application design. In this example, we choose Empty Activity.

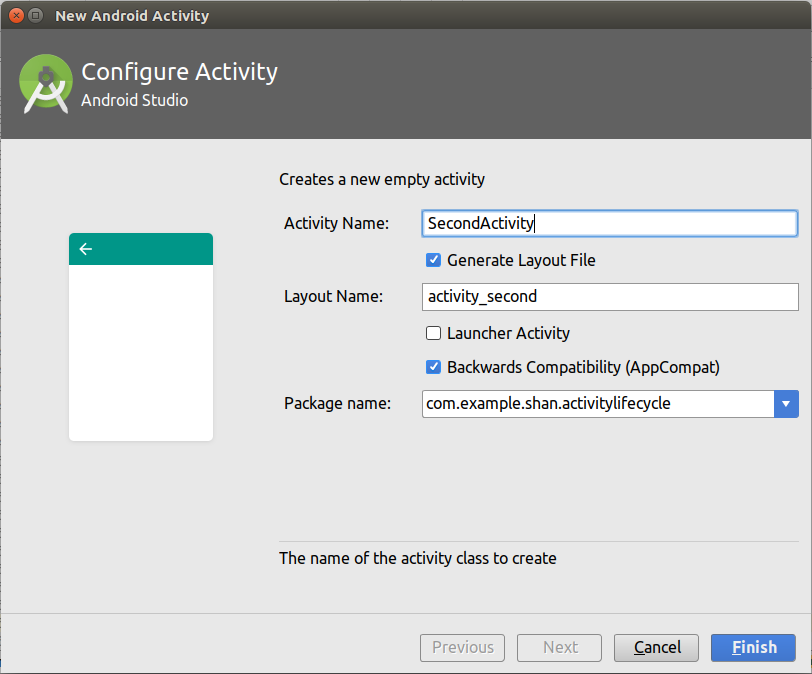


Figure 1: Creating new activity

After adding the new activity in the project as shown in Figure 1, in the project explorer, you can see the new activity class in the *java→ [package.name]*

Also, you can check the added XML layout file under project solution explorer as *res → layout → [name\_activity]*

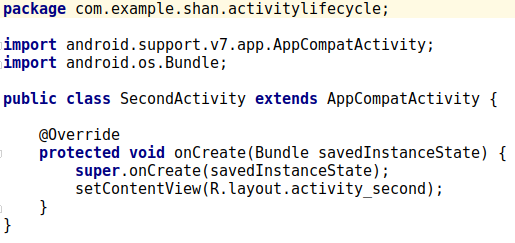


Figure 2: Main Activity class code

Figure 2 shows the auto generated code for the new activity class that extends AppCompatActivity class.

Activity class loads its user interface component using the XML file defined in your res/layout folder, using the function:

*setContentView(R.layout.activity\_second);*

The same activity is declared under AndroidManifest.xml as:



**Adding view element in the Activity:**

There are different view elements that can be added to activity UI to interact with users. Examples include Button, TextView, EditText and so on.

To add a new component in the UI, simply open the activity XML file, and on design tab, drag and drop view elements on the screen, as shown in the figure 3.

The properties of the these elements can be changed through Properties side panel in Android Studio after selecting the corresponding element.

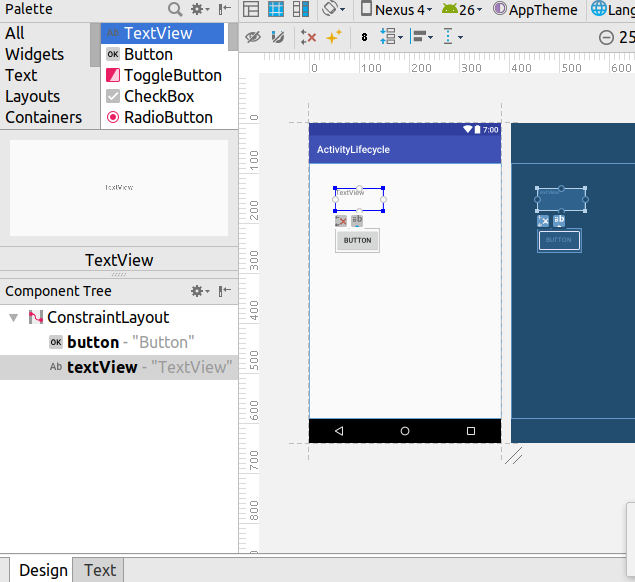


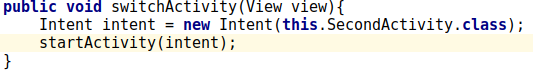
Figure 3: Adding views on the activity UI.

**Switching between Activities:**

Create two activities, MainActivity and SecondActivity.

Add a button on MainActivity and name it Switch Activity. Add a button on SecondActivity and name it Back.

Add the following in the callback handler of Switch Activity button in MainActivity to switch to second activity.



Also add the following XML property to the button XML in the layout file:

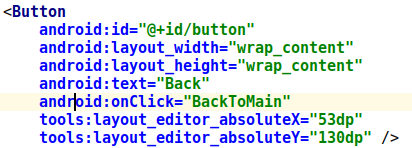
***android:onClick=switchActivity***

This is a must to register the onClick handler method name for this button.

Similarly, on the second activity, add the following code for Back button:



And remember to add the callback method name in the button’s xml as,



Note: Import the required class whenever needed. For instance, in the above examples, we need to import View and Intent classes as:

*import android.content.Intent;*

*import android.view.View;*

**ACTIVITIES**

**Activity 1**

Create a new application and add Login and Main activities in it. Add login button on Login activity and Back button on Main activity. On login button click, switch to Main activity, and on Back button click in Main activity, return to Login Activity.

**REVIEW QUESTIONS**

1. How do you add multiple activities in Android application?
2. What changes occur in AndroidManifest file when you add new activities?
3. What is the purpose of activity layout XML files?
4. How do you switch between activities?