BCS3283-Mobile Application Development

Chapter 4
Android Activity

Editor
Dr. Mohammed Falah Mohammed

Faculty of Computer Systems & Software Engineering
falah@ump.edu.my

Android Activity

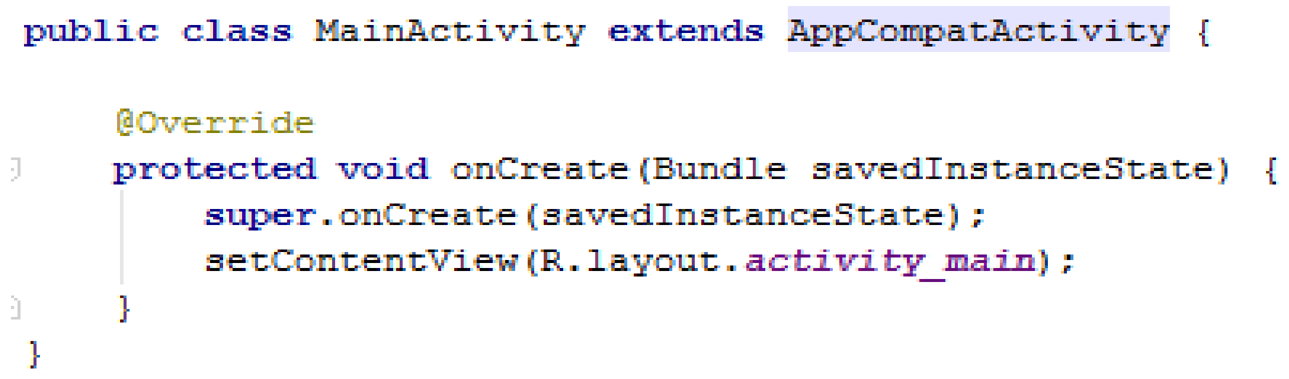
- Aims
  - To learn how to program a button in android studio.
- Expected Outcomes
  - Ability to create a button.
  - Ability to use setOnClickListener 1st method.
  - Ability to using onClick in Button property 2nd method
  - Build and Run Your App using Emulator and real device.
- References
  - https://developer.android.com

Android Activity

- An Activity is the code that works with a UI screen defined by the View.
- An Android app is made up of one or more Activities
- Activity layout is determined by:
  - the View- XML layout file contains a definition of the initial UI of your app (in res directory)
  - the behavior is determined by the Java code in the Java directory
- Similar like: a web page complete with HTML to determine what to displays, and JavaScript to determine what it does.

The Java

- There's more than one way to intercept the events from a user's interaction with your application.
- The most important method defined is onCreate.
- Activity class takes care of show the main screen and enable user to place the required UI.



```
public class MainActivity extends AppCompatActivity {

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

The Java

- All subclasses of Activity will implement based on two methods:
  - onCreate (bundle) : to initialize the activity and passing a Bundle object called savedInstanceState.
  - onPause() : to deal with the user leaving the activity.

- The MainActivity by default has no data (savedInstanceState is null), but user still have to pass it on to the inherited onCreate method.

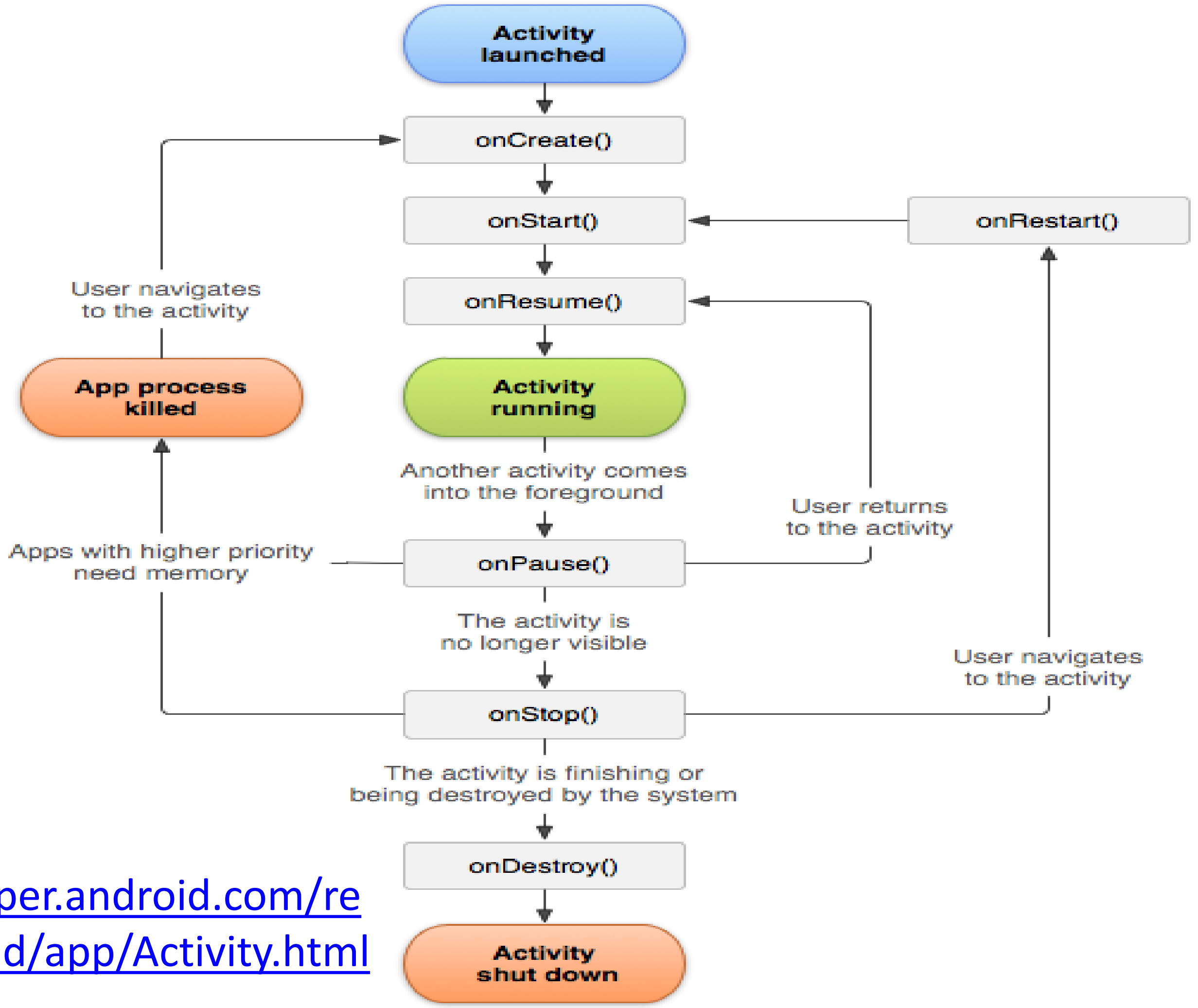
- setContentView : to pass an integer indicator to specify the described XML layout to be viewed.

The Java

- The R object (library object of resources) is constructed by the system,

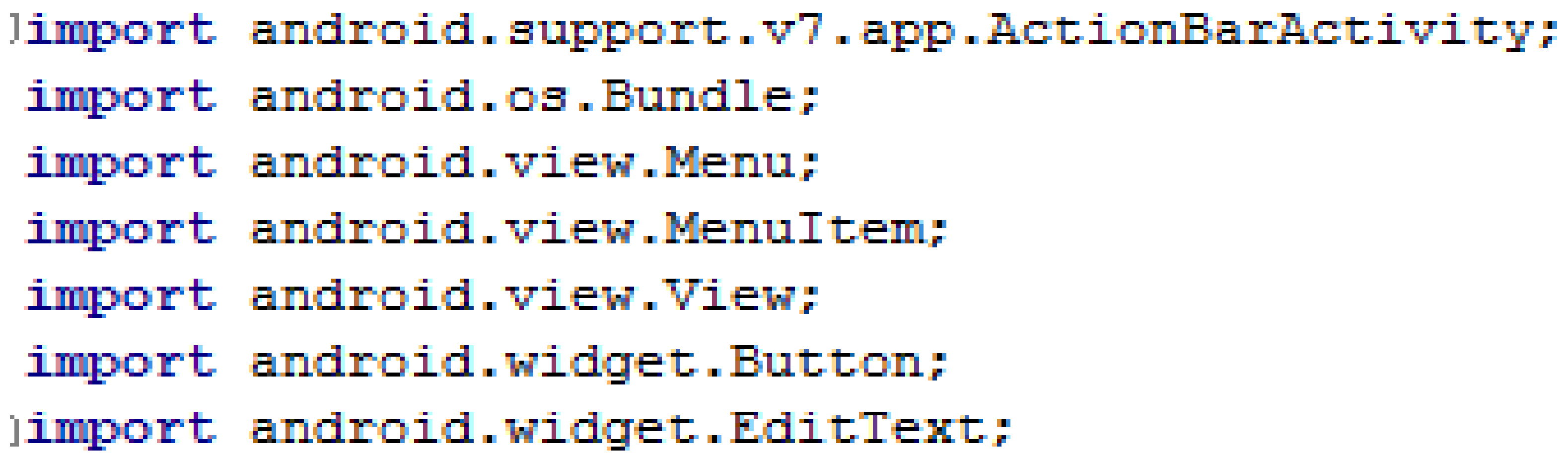
- where R.layout.activity_main allows the setContentView method to find the activity_main XML layout file by returns an integer value.

Activity state paths


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Source : https://developer.android.com/reference/android/app/Activity.html

Import libraries

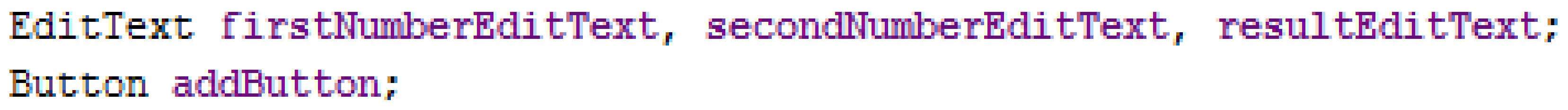


```
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

- Automatically generated once programmer type the words Button and EditText during object declarations.

- Instead of ActionBarActivity API version 23 will **import** AppCompatActivity.

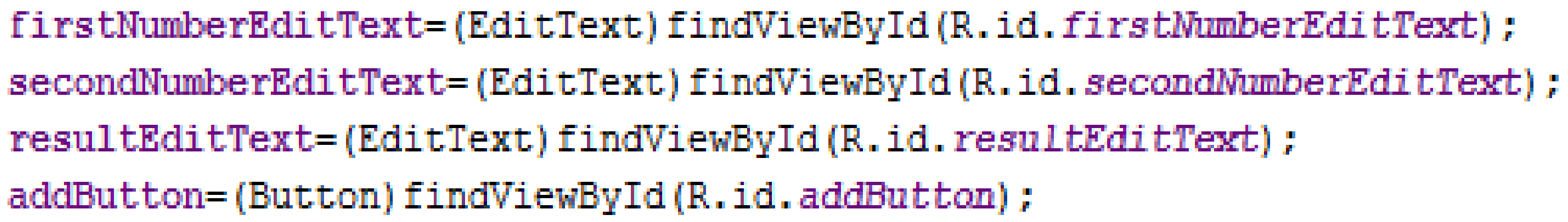
Object declaration



```
EditText firstNumberEditText, secondNumberEditText, resultEditText;
Button addButton;
```

- Not all widgets have to be declared. Depends on your apps functionality.
- If certain Value of the widgets will be change during the life cycle of the apps, then you have to declare.
- If the value is static (not change) then no need to put inside of the java file.

Link Java Objects with widgets



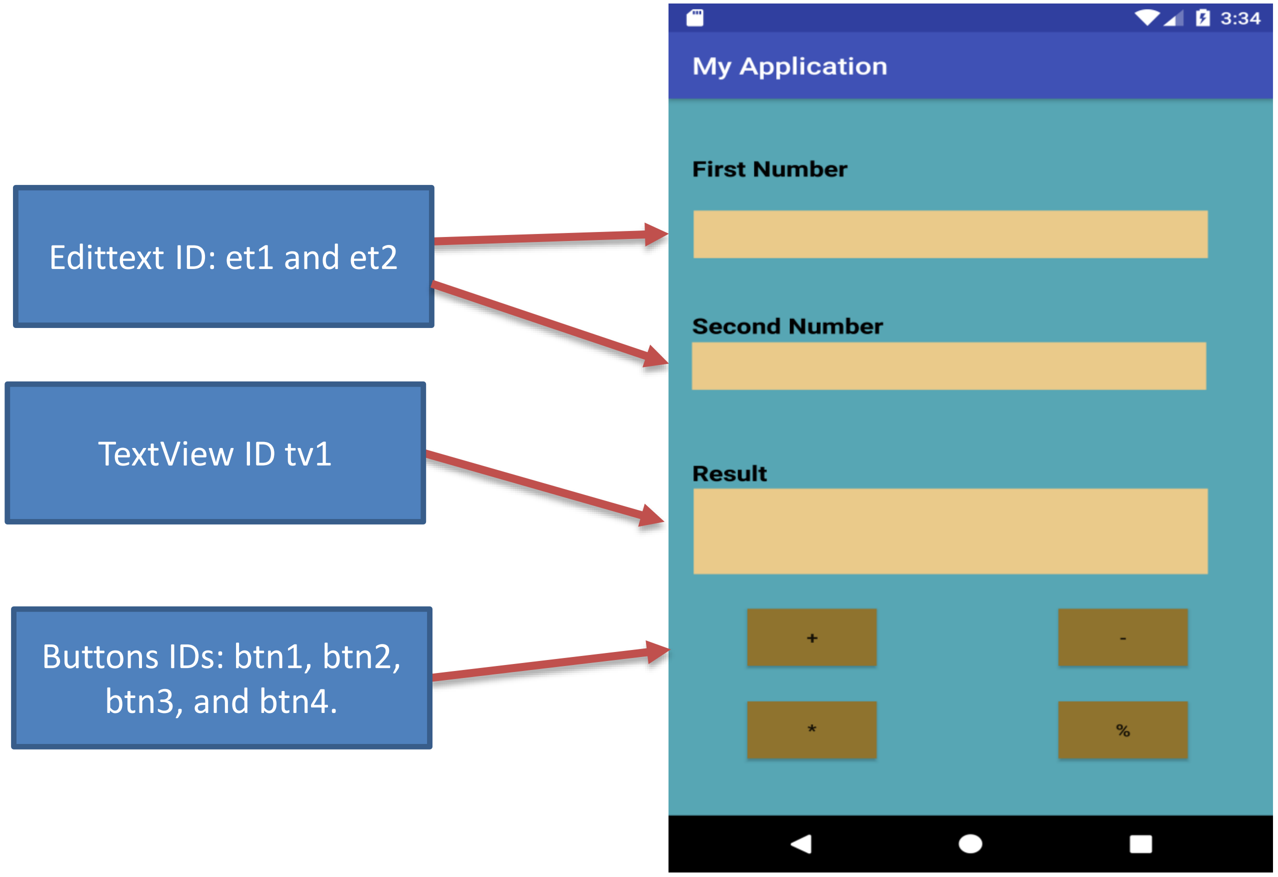
```
firstNumberEditText=(EditText)findViewById(R.id.firstNumberEditText);
secondNumberEditText=(EditText)findViewById(R.id.secondNumberEditText);
resultEditText=(EditText)findViewById(R.id.resultEditText);
addButton=(Button)findViewById(R.id.addButton);
```

- It is important for you to change the id of each view (layout/widgets/etc..) inside
  the xml file corresponding for the GUI design.

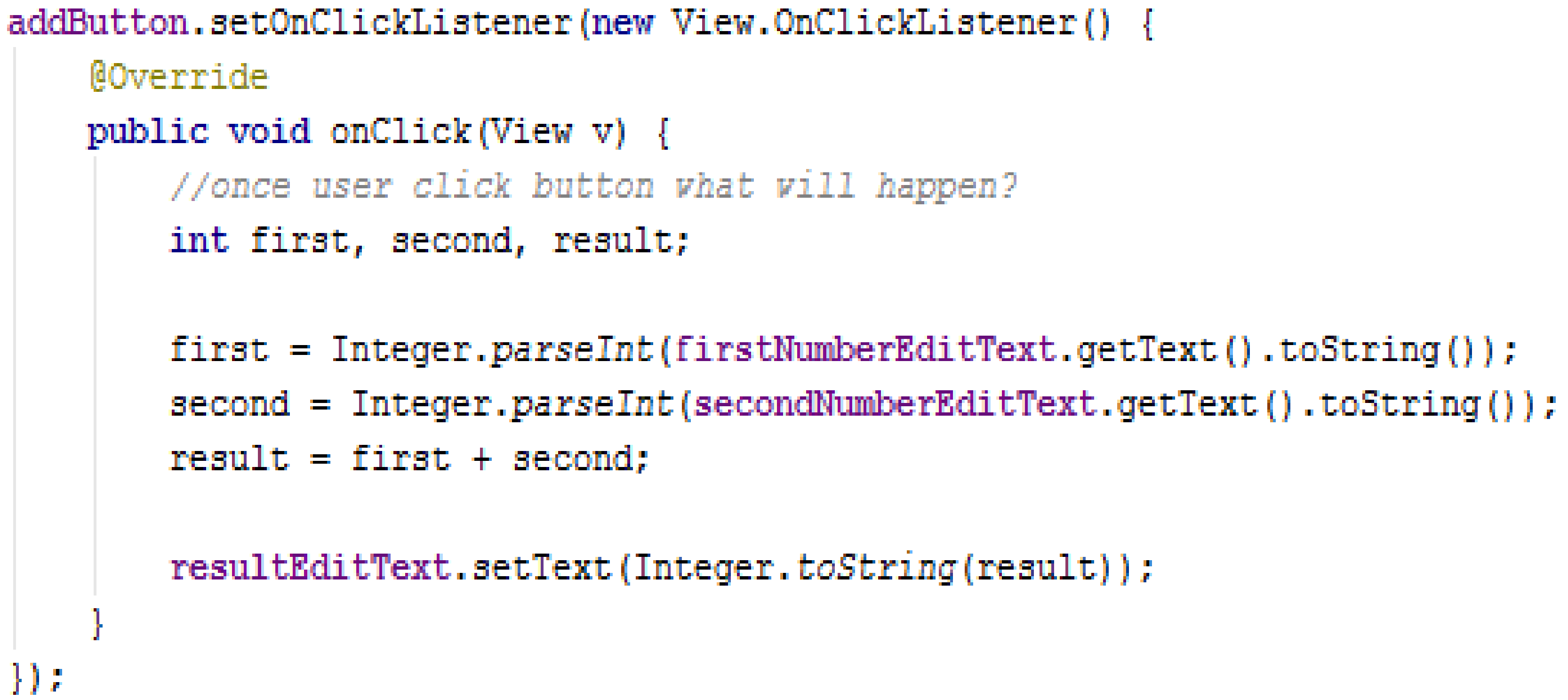
- This id will be used to link between the widgets in xml file with the widget objects declared inside of java file.

Simple Calculator App

- Create new application with GUI:



Program your button: 1st method

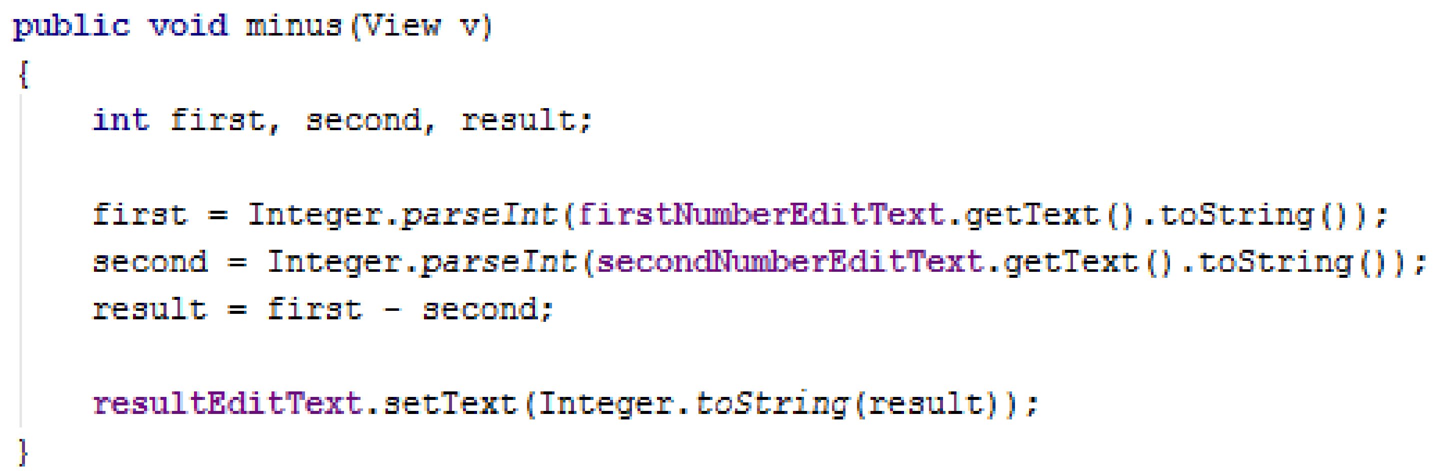


```
addButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //once user click button what will happen?
        int first, second, result;

        first = Integer.parseInt(firstNumberEditText.getText().toString());
        second = Integer.parseInt(secondNumberEditText.getText().toString());
        result = first + second;

        resultEditText.setText(Integer.toString(result));
    }
});
```

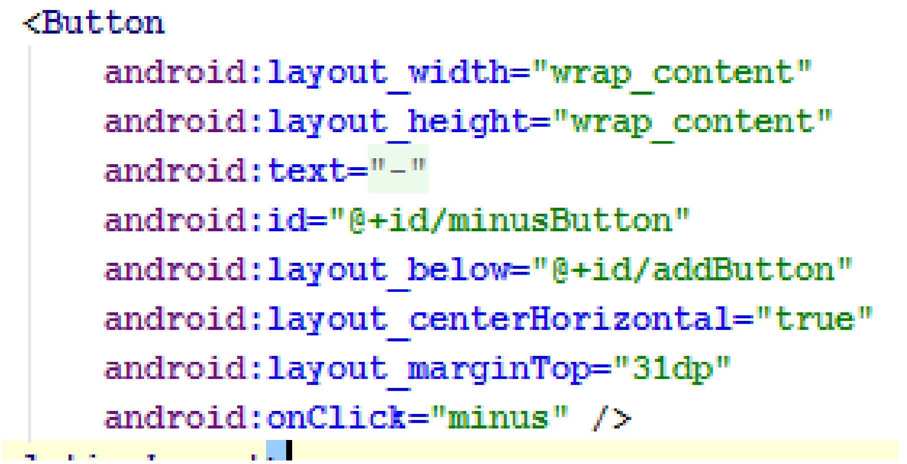
Program your button: 2nd method



```java
public void minus(View v)
{
    int first, second, result;

    first = Integer.parseInt(firstNumberEditText.getText().toString());
    second = Integer.parseInt(secondNumberEditText.getText().toString());
    result = first - second;

    resultEditText.setText(Integer.toString(result));
}
```



```xml
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="-"
    android:id="@+id/minusButton"
    android:layout_below="@+id/addButton"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="31dp"
    android:onClick="minus" />
```