ANDROID TUTORIAL

L - 5

3rd Sem, MCA

LAB-4

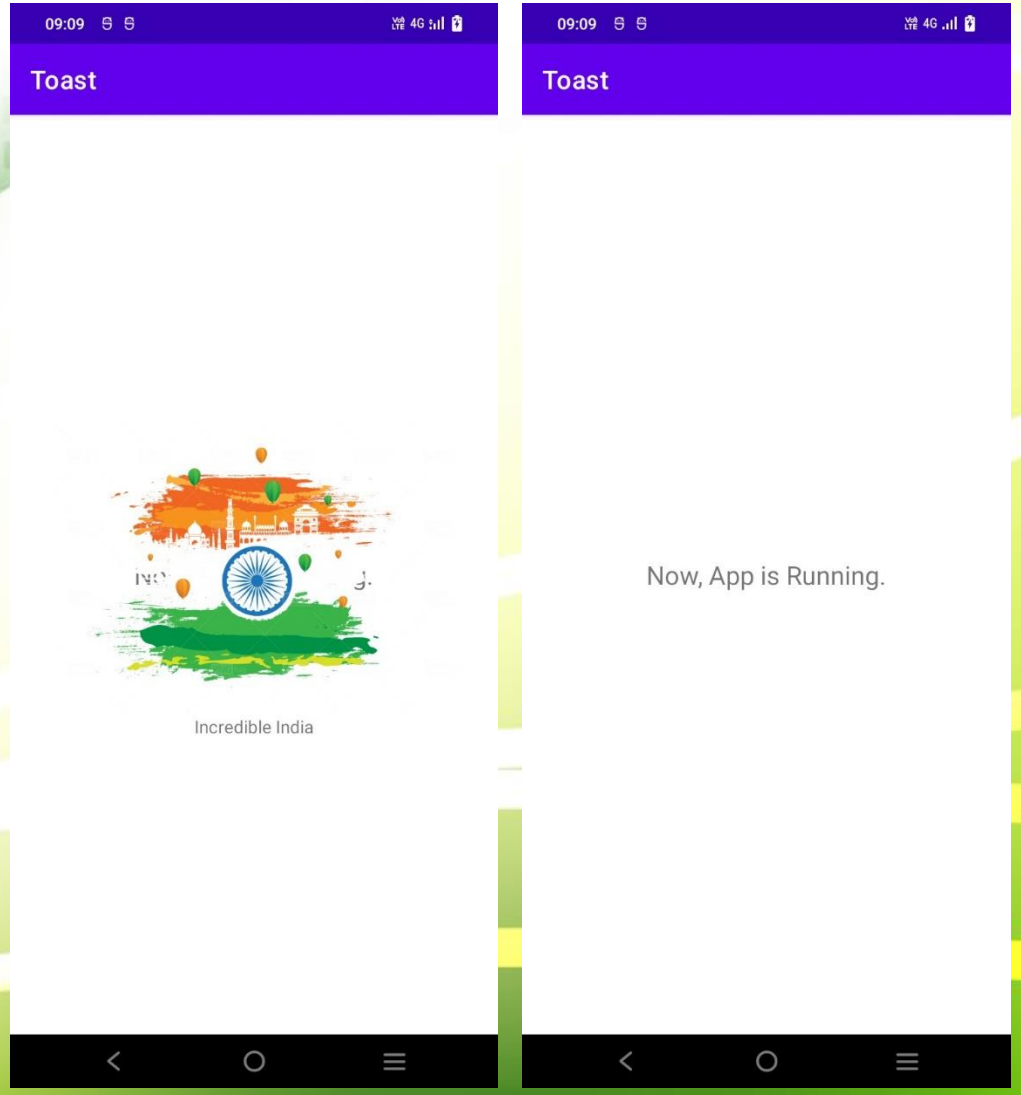
L4	7. For the Project in L3/T5; add the following features; a) Display the info in DIFFERENT page/activity. b) Enable 'back' button to return to Home page. c) Use toggle button to 'Reset' input area. 8. Create a **Master App** by integrating ALL previous programs into one single App. Provide suitable options to navigate between activities.

LAB-5

L5	9. Create an android application to display a list of bank names in a spinner and when you select the particular bank name, the logo and IFSC-code of the selected bank should be displayed using custom toast. 10. Create an android application to depict the option menu with following features: a) Toast menu option to display a Toast message. b) Exit menu to close the App. c) President menu should open a new activity with two fragments. First fragment should contain a listview of all presidents of Independent India. d) When you select any president's name, corresponding details should be visible in the second fragment. e) Details should include: Name, Presidential period, Qualification, Professional experience, Lifetime, Special Achievements, & Image, etc.

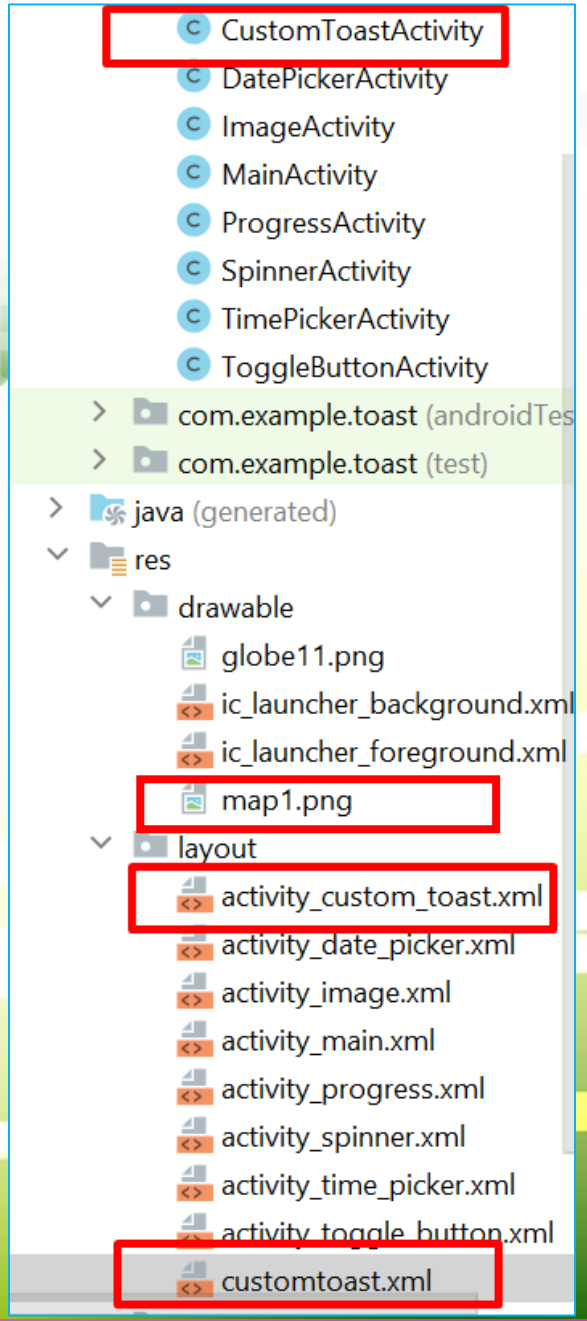
CUSTOM TOAST

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CustomToastActivity">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:text="Now, App is Running."
    android:textSize="20dp"></TextView>
</RelativeLayout>
```



CUSTOM TOAST

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/a
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/custom_toast_layout"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/custom_toast_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:src="@drawable/map1" />
    <TextView
        android:id="@+id/custom_toast_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="420dp"
        android:text="Incredible India" />
```



CUSTOM TOAST

LayoutInflater class is used to instantiate the contents of layout XML files into their corresponding View objects

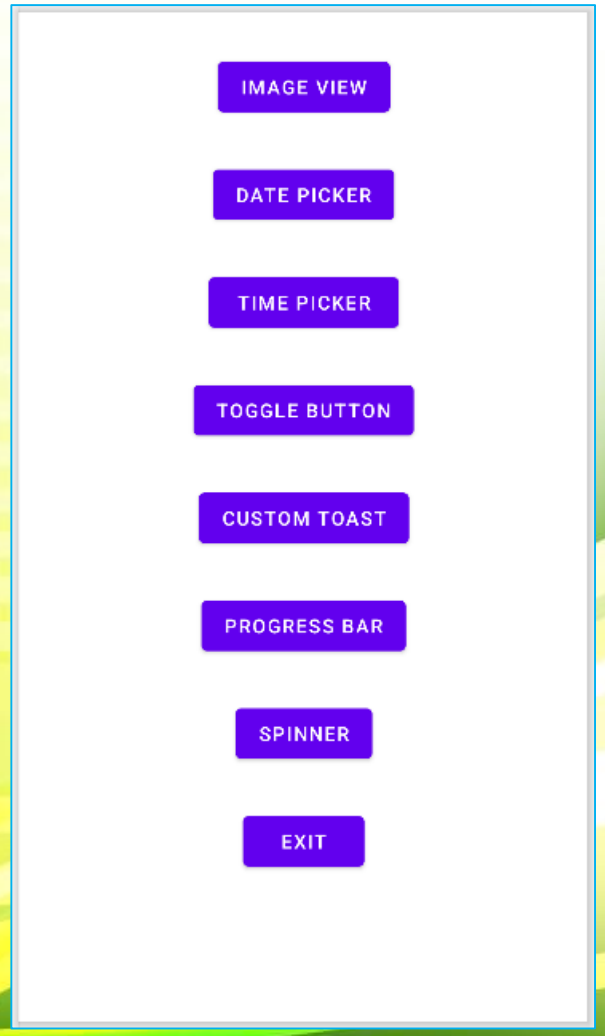
→ it takes an XML file as input and builds the View objects from it.

```
//Creating the LayoutInflater instance
LayoutInflater li = getLayoutInflater();

//Getting the View object as defined in the customtoast.xml file
View layout = li.inflate(R.layout.customtoast,(ViewGroup) findViewById(R.id.custom_toast_layout));

//Creating the Toast object
Toast toast = new Toast(getApplicationContext());
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);      //setting the view of custom toast layout
toast.show();
```

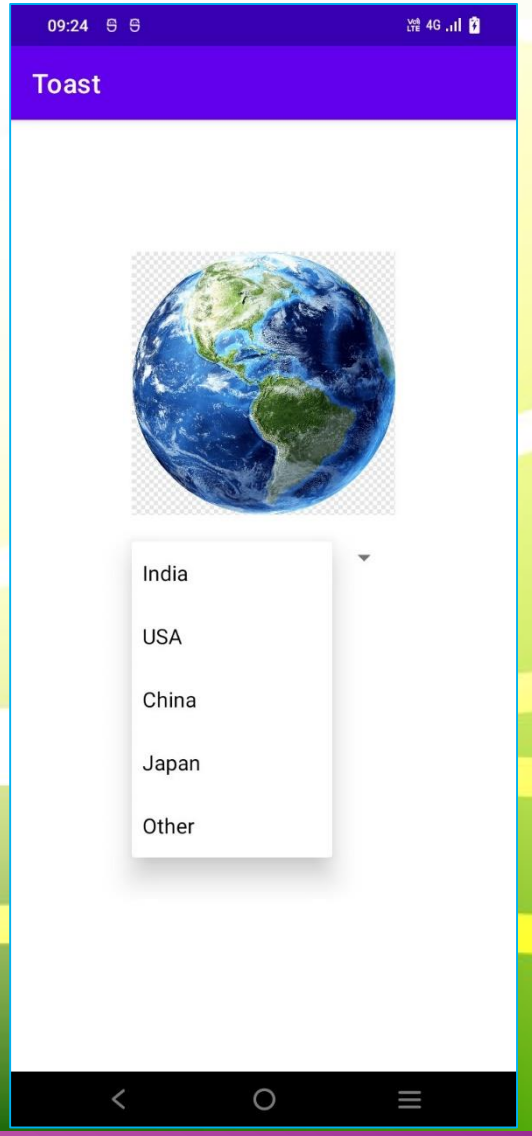
EXIT




```
btnExit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        MainActivity.this.finish();
        System.exit( status: 0);
    }
});
```

SPINNER

- Spinner is a view that allows user to select one value from list of values.
- Behave same as a dropdown list in other programming languages.
- Spinner control with list of choices can be populated by defining an ArrayAdapter in Activity file.
- **Adapter** acts as a bridge between the UI Component and the Data Source.
- Converts data from data sources into view items that can be displayed into the UI Component.
- Data Source can be Arrays, HashMap, Database, etc. and UI Components can be ListView, GridView, Spinner, etc.
- **ArrayAdapter** is the most commonly used adapter in android.



SPINNER

```
public class SpinnerActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

    String[] country = { "India", "USA", "China", "Japan", "Other"};

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

        //Getting the instance of Spinner and applying OnItemSelectedListener on it
        Spinner spin = (Spinner) findViewById(R.id.spinner);

        //Creating the ArrayAdapter instance having the country list
        ArrayAdapter aa = new ArrayAdapter( context: this,android.R.layout.simple_spinner_item,country);
        aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        //Setting the ArrayAdapter data on the Spinner
        spin.setAdapter(aa);

        spin.setOnItemSelectedListener(this);
    }
```

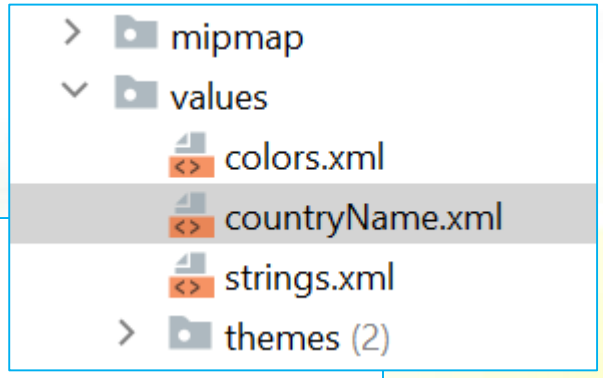
SPINNER

```java
//Performing action onItemSelected and onNothing selected
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1, int position, long id) {
    Toast.makeText(getApplicationContext(),country[position] , Toast.LENGTH_SHORT).show();
}

@Override
public void onNothingSelected(AdapterView<?> arg0) {
    // TODO Auto-generated method stub
}
```

```xml
<Spinner
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"
    android:minWidth="200dp"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:layout_below="@id/imageView"/>
```

SPINNER




```
//Creating the ArrayAdapter instance having the country list
ArrayAdapter aa = ArrayAdapter.createFromResource( context: this,
        R.array.state_array, android.R.layout.simple_spinner_item);
//  ArrayAdapter aa = new ArrayAdapter(this,android.R.layout.simple_spinner_item,country);
aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

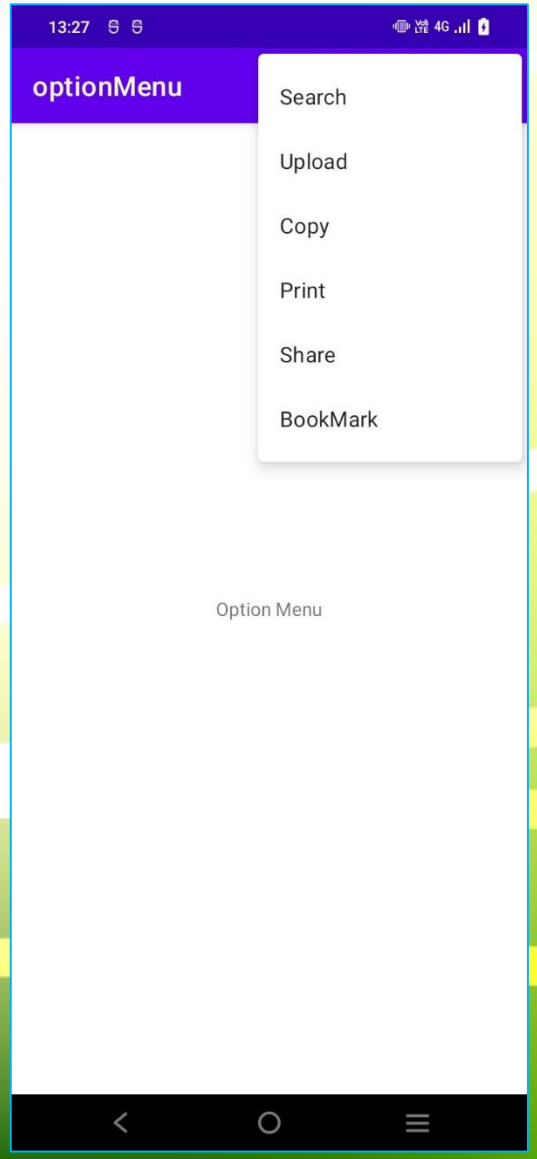
```
String[] strState = getResources().getStringArray(R.array.state_array);
ArrayList arrState = new ArrayList(Arrays.asList(strState));
ArrayAdapter aa = new ArrayAdapter( context: this,android.R.layout.simple_spinner_item,arrState);
```

```
ArrayList<String> state = new ArrayList<String>();
state.add("Bihar");
state.add("Manipur");
state.add("Sikkim");
ArrayAdapter aa = new ArrayAdapter( context: this,android.R.layout.simple_spinner_item,state);
```

```
ml version="1.0" encoding="utf-8"?>
sources>
 <string-array name="state_array">
     <item>Karnataka</item>
     <item>Goa</item>
     <item>Punjab</item>
     <item>Assam</item>
 </string-array>
</resources>
```

MENUS

- In android, Menu is a part of the user interface (UI) component which is used to handle some common functionality around the application.
- By using Menus in our applications, we can provide better and consistent user experience throughout the application.
- In android, we can define a Menu in separate XML file and use that file in our activities or fragments based on our requirements.

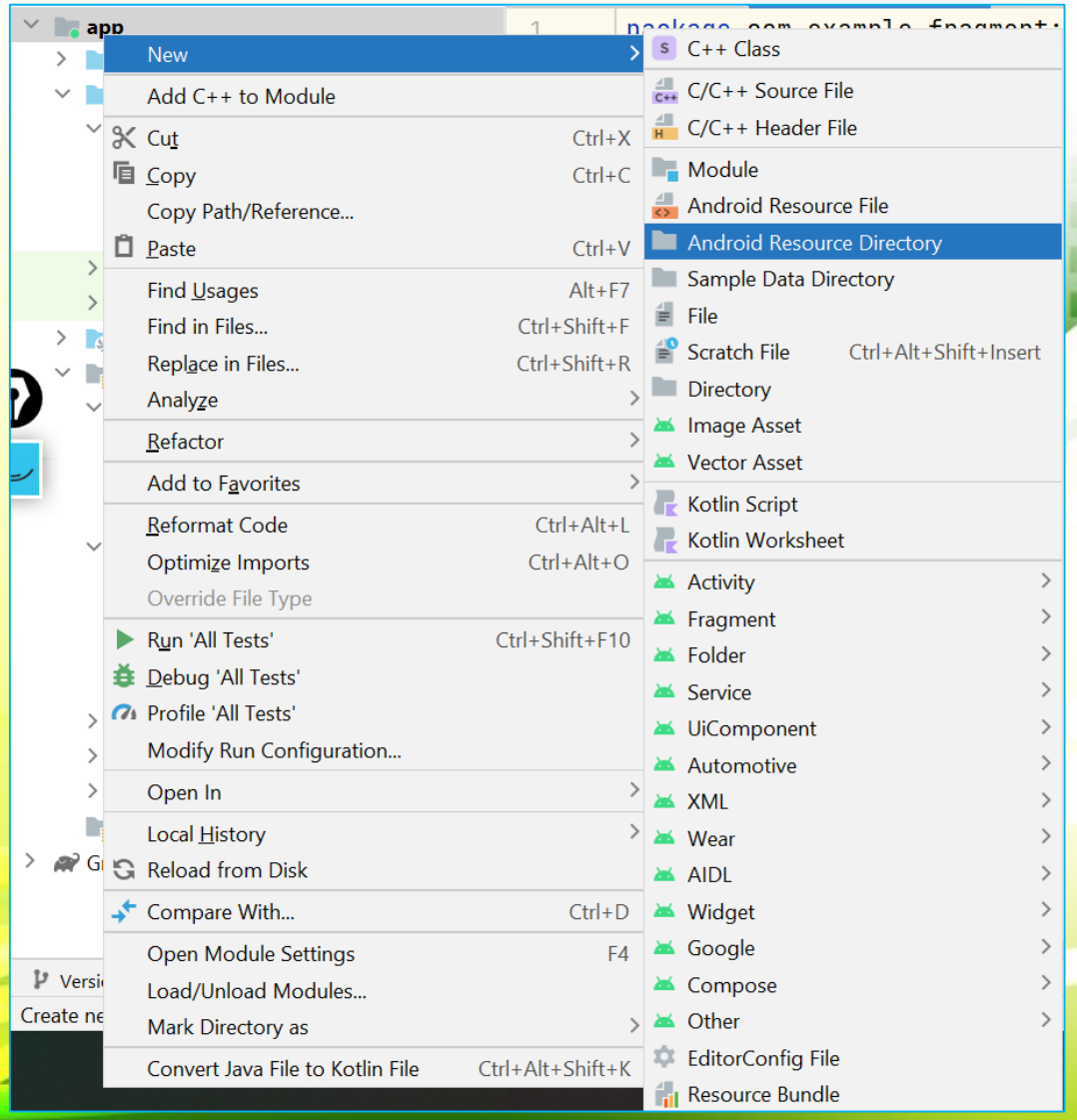


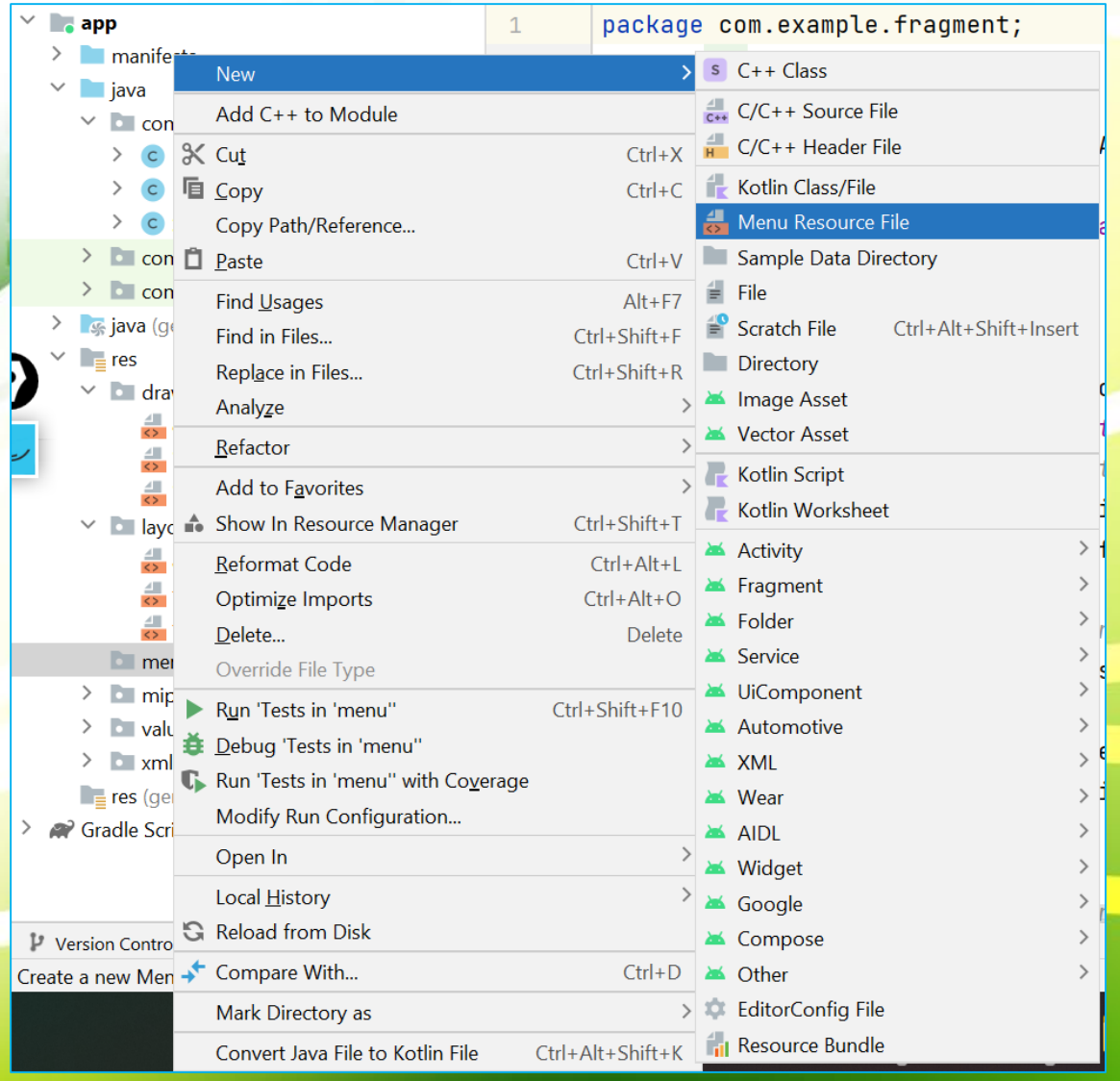
MENUS

- For all menu types, Android provides a standard XML format to define menu items.
- Instead of building a menu in our activity's code, we should define a menu and all its items in an XML menu resource and load menu resource as a Menu object in our activity or fragment.
- In android, to define menu, we need to create a new folder menu inside of our project resource directory (res/menu/) and add a new XML file to build the menu with the following elements.

Element	Description
<menu>	It's a root element to define a Menu in XML file and it will hold one or more and elements.
<item>	It is used to create a menu item and it represents a single item on the menu. This element may contain a nested <menu> element in order to create a submenu.
<group>	It's an optional and invisible for <item> elements. It is used to categorize the menu items so they share properties such as active state and visibility.

MENUS


app
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Override File Type
Run 'All Tests' Ctrl+Shift+F10
Debug 'All Tests'
Profile 'All Tests'
Modify Run Configuration...
Open In
Local History
Reload from Disk
Compare With... Ctrl+D
Open Module Settings F4
Load/Unload Modules...
Mark Directory as
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
C++ Class
C/C++ Source File
C/C++ Header File
Module
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Compose
Other
EditorConfig File
Resource Bundle


app
1
package com.example.fragment;
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show In Resource Manager Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Override File Type
Run 'Tests in 'menu'' Ctrl+Shift+F10
Debug 'Tests in 'menu''
Run 'Tests in 'menu'' with Coverage
Modify Run Configuration...
Open In
Local History
Reload from Disk
Compare With... Ctrl+D
Mark Directory as
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
C++ Class
C/C++ Source File
C/C++ Header File
Kotlin Class/File
Menu Resource File
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Compose
Other
EditorConfig File
Resource Bundle

MENUS

Attribute	Description
android: id	It is used to uniquely identify an element in the application.
android:icon	It is used to set the item's icon from drawable folder.
android: title	It is used to set the item's title
android:showAsAction	It is used to specify how the item should appear as an action item in the app bar.

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/search_item"
        android:title="Search" />
    <item android:id="@+id/upload_item"
        android:title="Upload" />
    <item android:id="@+id/copy_item"
        android:title="Copy" />
    <item android:id="@+id/print_item"
        android:title="Print" />
    <item android:id="@+id/share_item"
        android:title="Share" />
    <item android:id="@+id/bookmark_item"
        android:title="BookMark" />
</menu>
```

MENU

```java
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.options_menu, menu);
    return true;
}
```

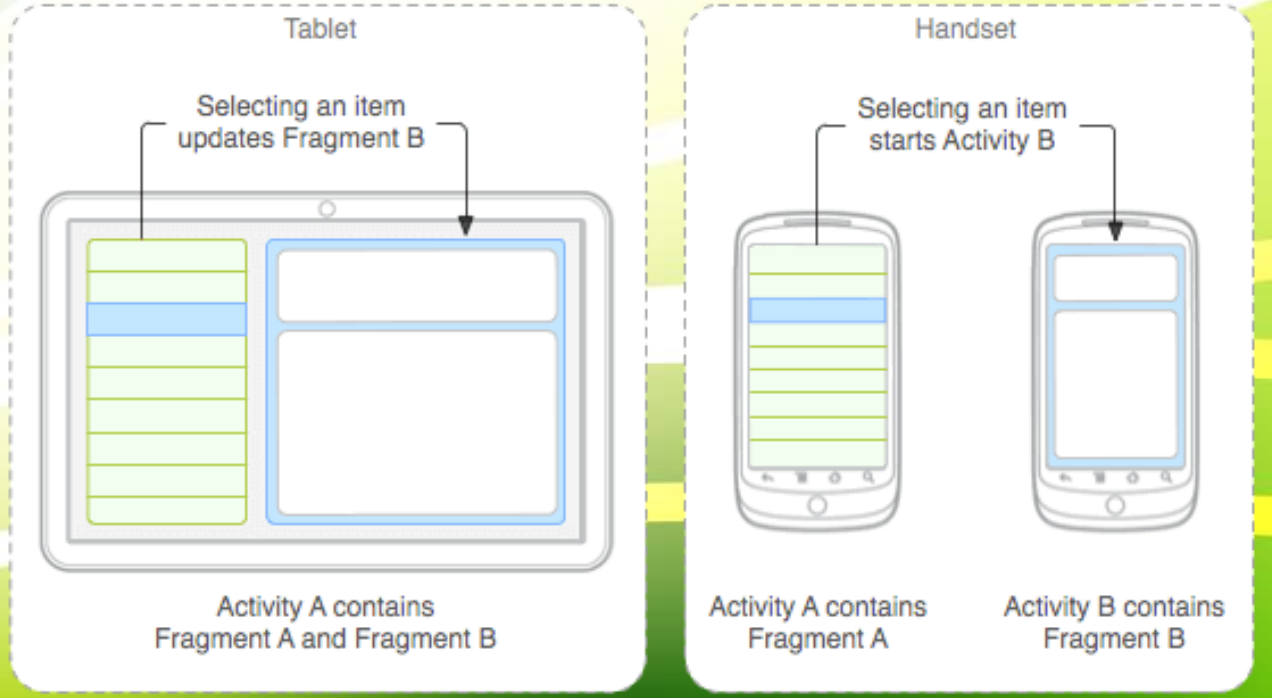
```xml
<item android:id="@+id/upload_item"
    android:title="Upload" >
    <menu>
        <item android:id="@+id/search_item2"
            android:title="Search" />
    </menu>
    </item>
<item android:id="@+id/copy_item"
    android:title="Copy" />
<item android:id="@+id/print_item"
    android:title="Print" />
```

MENU CLICK EVENTS

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Toast.makeText( context: this,  text: "Selected Item: " +item.getTitle(),
    switch (item.getItemId()) {
        case R.id.search_item:
            // do your code
            return true;
        case R.id.upload_item:
            // do your code
            return true;
        case R.id.copy_item:
            // do your code
            return true;
        case R.id.print_item:
            // do your code
            return true;
        case R.id.share_item:
            // do your code
            return true;
```

FRAGMENTS

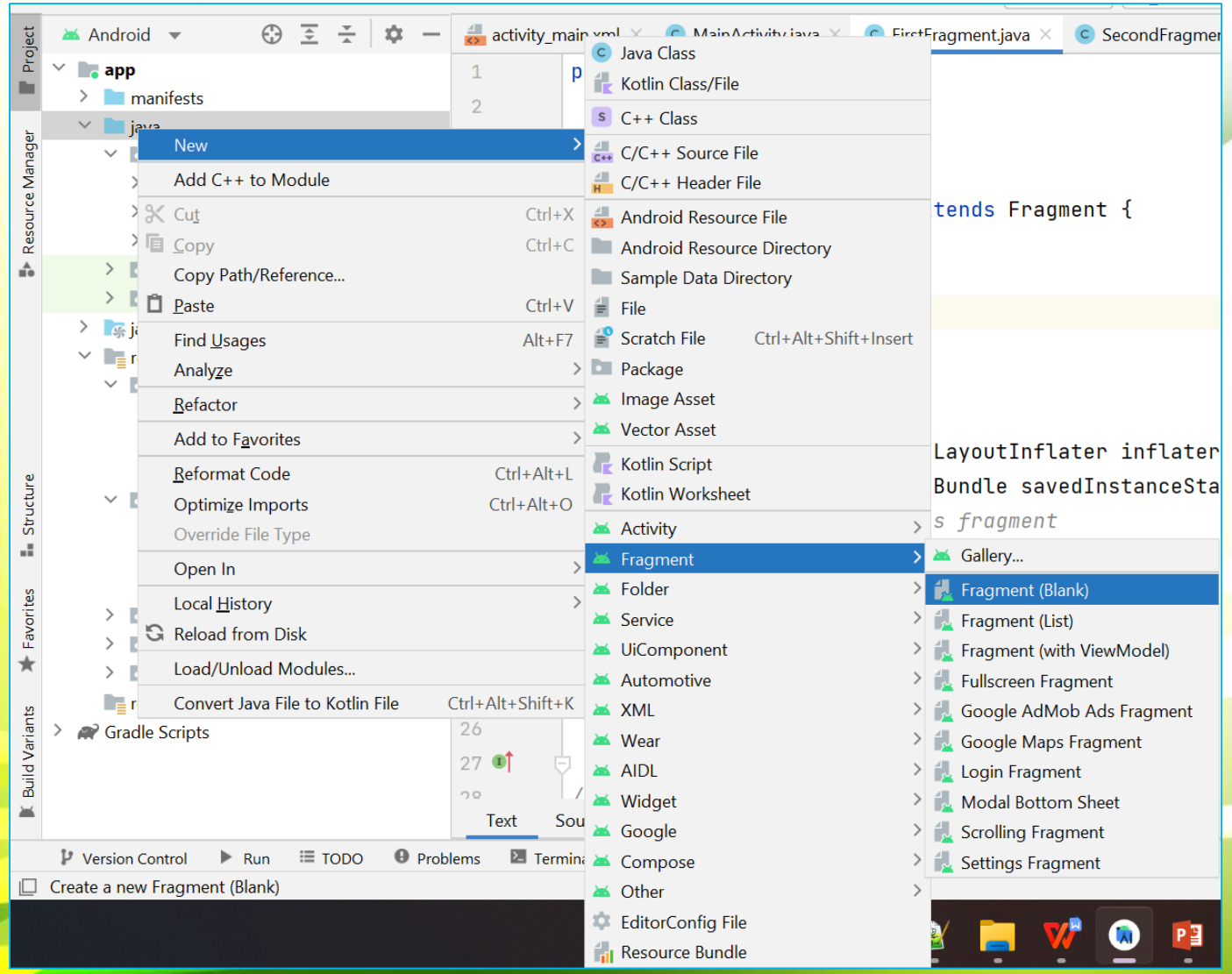
- Fragments are the modular section of activity design, and used to represent UI behavior in an activity.
- Fragments used to create flexible UI designs that can be adjusted based on the device screen size such as tablets, smartphones.
- We can build multi-pane UI by combining multiple fragments in a single activity and we can reuse the same fragment in multiple activities.
- Fragment has its own lifecycle call-backs and accepts its own input events.
- We can add or remove fragments in an activity while the activity is running.
- Fragment will act as a sub-activity and we can reuse it in multiple activities.
- We can insert fragment into activity layout by using <fragment> element and by dividing the layout of activity into fragments, we can modify the appearance of an app design at runtime.



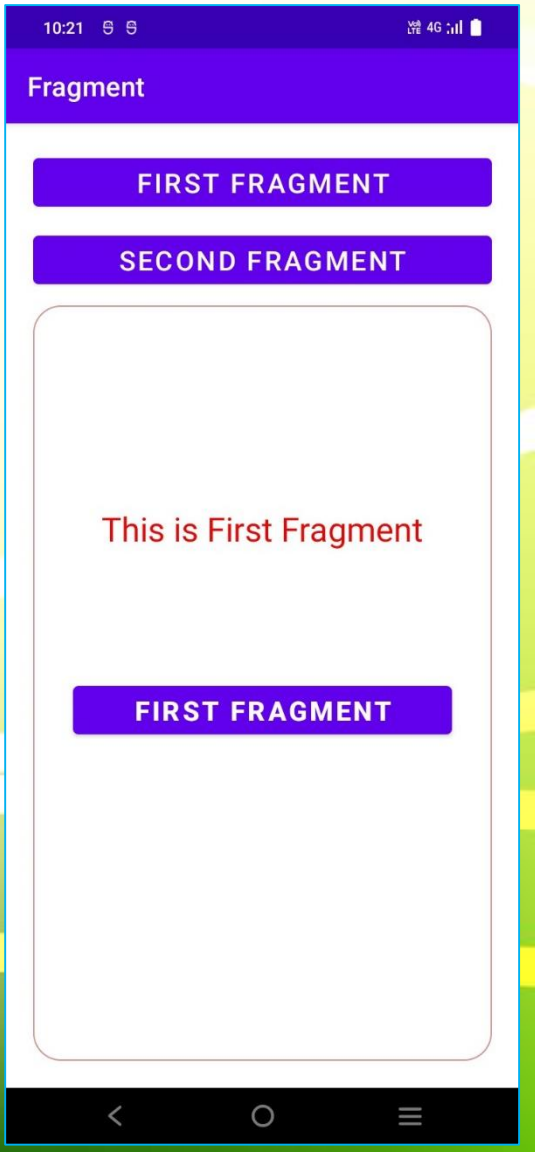
ANDROID FRAGMENT LIFE CYCLE

Method	Description
onAttach()	It is called when the fragment has been associated with an activity.
onCreate()	It is used to initialize the fragment.
onCreateView()	It is used to create a view hierarchy associated with the fragment.
onActivityCreated()	It is called when the fragment activity has been created and the fragment view hierarchy instantiated.
onStart()	It is used to make the fragment visible.
onResume()	It is used to make the fragment visible in an activity.
onPause()	It is called when fragment is no longer visible and it indicates that the user is leaving the fragment.
onStop()	It is called to stop the fragment using the onStop() method.
onDestoryView()	The view hierarchy associated with the fragment is being removed after executing this method.
onDestroy()	It is called to perform a final clean up of the fragments state.
onDetach()	It is called immediately after the fragment disassociated from the activity.

FRAGMENT


Android
app
manifests
java
Project
Resource Manager
Structure
Favorites
Build Variants
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Override File Type
Open In
Local History
Reload from Disk
Load/Unload Modules...
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
Gradle Scripts
Java Class
Kotlin Class/File
C++ Class
C/C++ Source File
C/C++ Header File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Compose
Other
EditorConfig File
Resource Bundle
Gallery...
Fragment (Blank)
Fragment (List)
Fragment (with ViewModel)
Fullscreen Fragment
Google AdMob Ads Fragment
Google Maps Fragment
Login Fragment
Modal Bottom Sheet
Scrolling Fragment
Settings Fragment
tends Fragment {
LayoutInflater inflater
Bundle savedInstanceSta
s fragment
Text
Version Control
Run
TODO
Problems
Create a new Fragment (Blank)

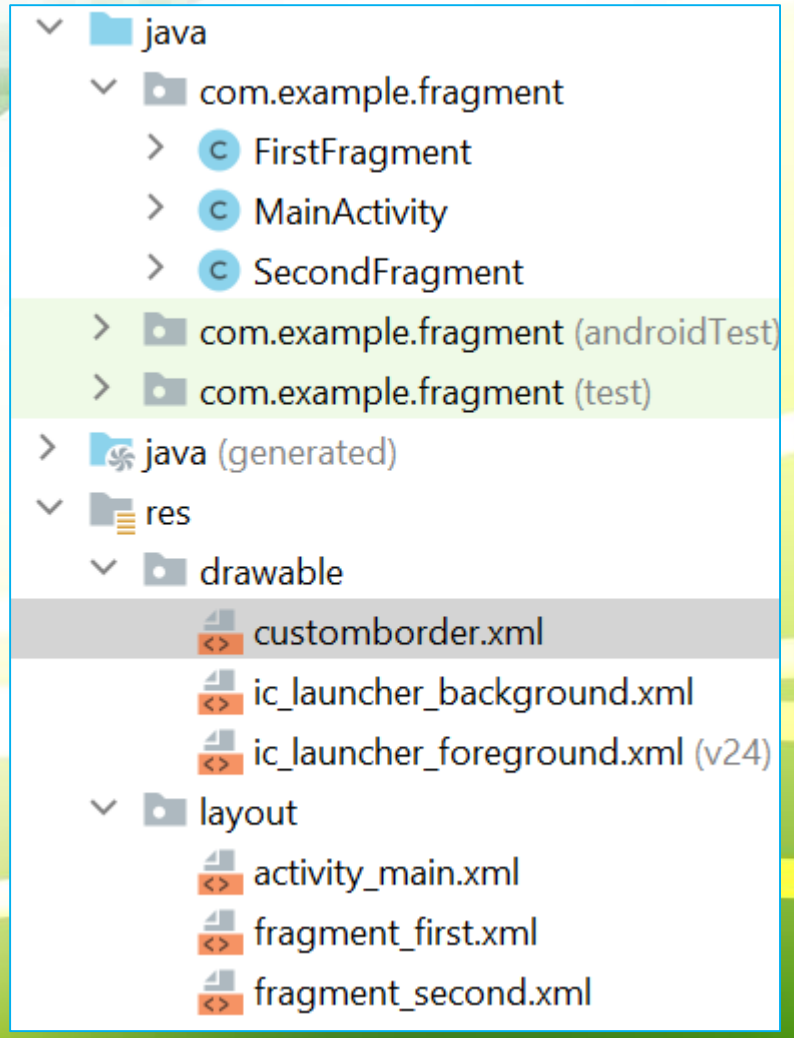

10:21
Fragment
FIRST FRAGMENT
SECOND FRAGMENT


10:21
Fragment
FIRST FRAGMENT
SECOND FRAGMENT
This is First Fragment
FIRST FRAGMENT

FRAGMENT

```
<FrameLayout
    android:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="10dp"
    android:paddingTop="50dp"
    android:background="@drawable/customborder"/>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="20dp"/>
    <padding android:left="10dp"
        android:right="10dp"
        android:top="10dp"
        android:bottom="10dp"/>
    <stroke android:width="1dp"
        android:color="#CF9494"/>
</shape>
```



FRAGMENT

```
public class FirstFragment extends Fragment {
    View view;
    Button firstButton;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
// Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_first, container,  attachToRoot: false);
// get the reference of Button
        firstButton = (Button) view.findViewById(R.id.firstButton);
// perform setOnClickListener on first Button
        firstButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
// display a message by using a Toast
                Toast.makeText(getActivity(),  text: "First Fragment", Toast.LENGTH_LONG).show();
            }
        });
        return view;
```

FRAGMENT

```
// perform setOnClickListener event on First Button
firstFragment.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        loadFragment(new FirstFragment());      // load First Fragment
    }
});
```

```
private void loadFragment(Fragment fragment) {
    // create a FragmentManager
    FragmentManager fm = getFragmentManager();
    // create a FragmentTransaction to begin transaction and replace Fragment
    FragmentTransaction fragmentTransaction = fm.beginTransaction();
    // replace the FrameLayout with new Fragment
    fragmentTransaction.replace(R.id.frameLayout, fragment);
    fragmentTransaction.commit(); // save the changes
}
```