SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Year 2

IT2010 - Mobile Application Design and Development

Lab sheet 04

Android Fragments

What is a Fragment?

- A reusable class implementing portion of an Activity.
- Defines a part of a User Interface.
- Must be embedded in activities, they cannot rum independently of activities.
- A combination of an XML layout file and a java class much like an Activity.
- Encapsulate views and logic. So that it is easier to reuse within activities.
- Standalone components that can contain views, events and logic.

Importance of Fragments

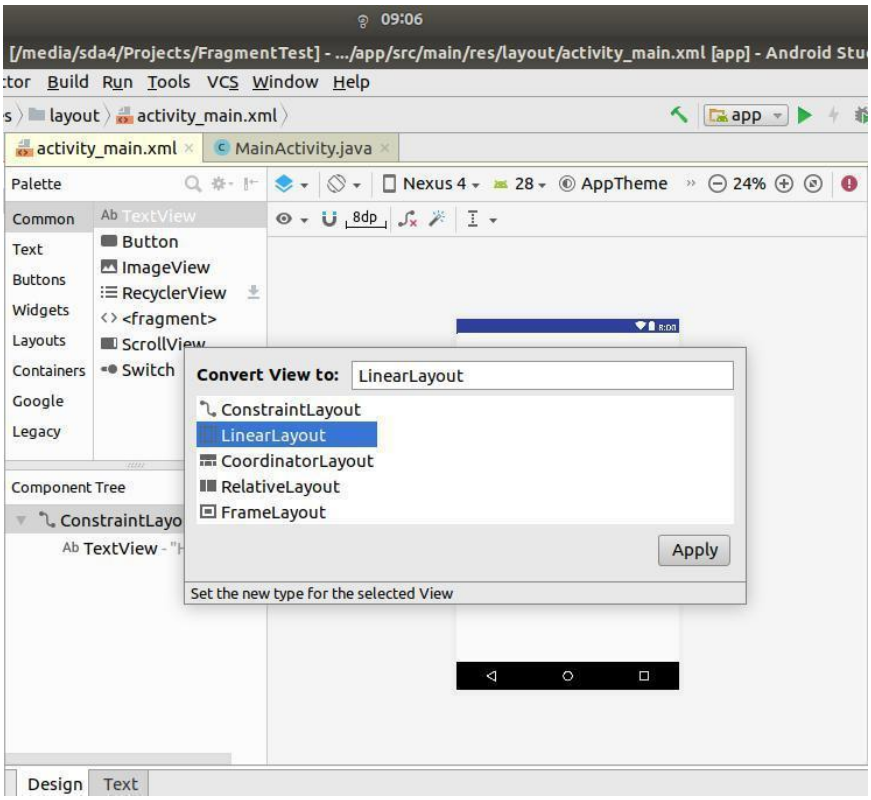
- Reusing View and Logic Components
- Tablet Support
- Screen Orientation

Embedding a Fragment in an Activity
There are two ways to add a fragment to an Activity.

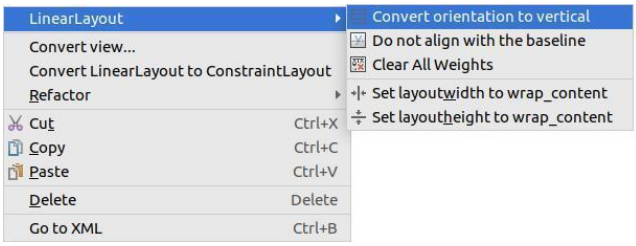
1) Dynamically using Java
2) Statically using XML

NOTE: The Activity should extend from FragmentActivity or AppCompactActivity.

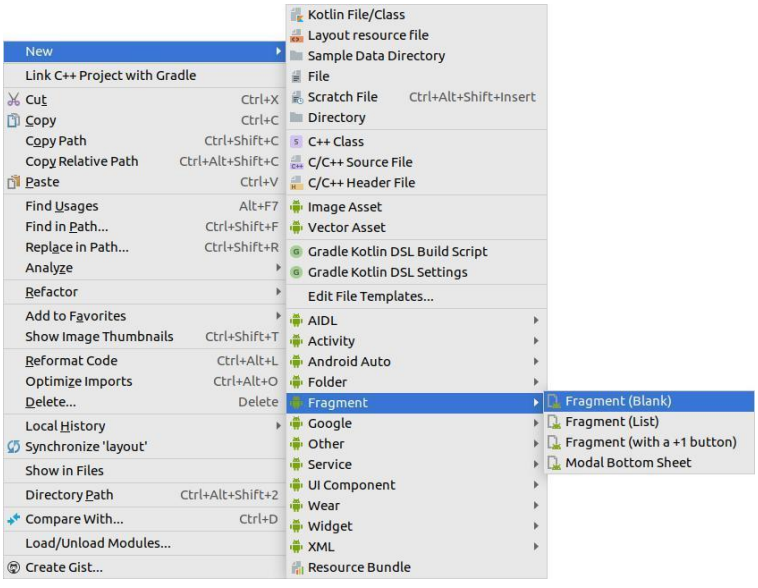
1. Start a project name “AndroidFragments”.
2. Change the layout from Constraint Layout to Linear layout.
   a) Go to activity_main.xml
   b) Right Click “ConstraintLayout” from the component tree and click “Convert View ...”
   c) Select “LinearLayout” and click apply.



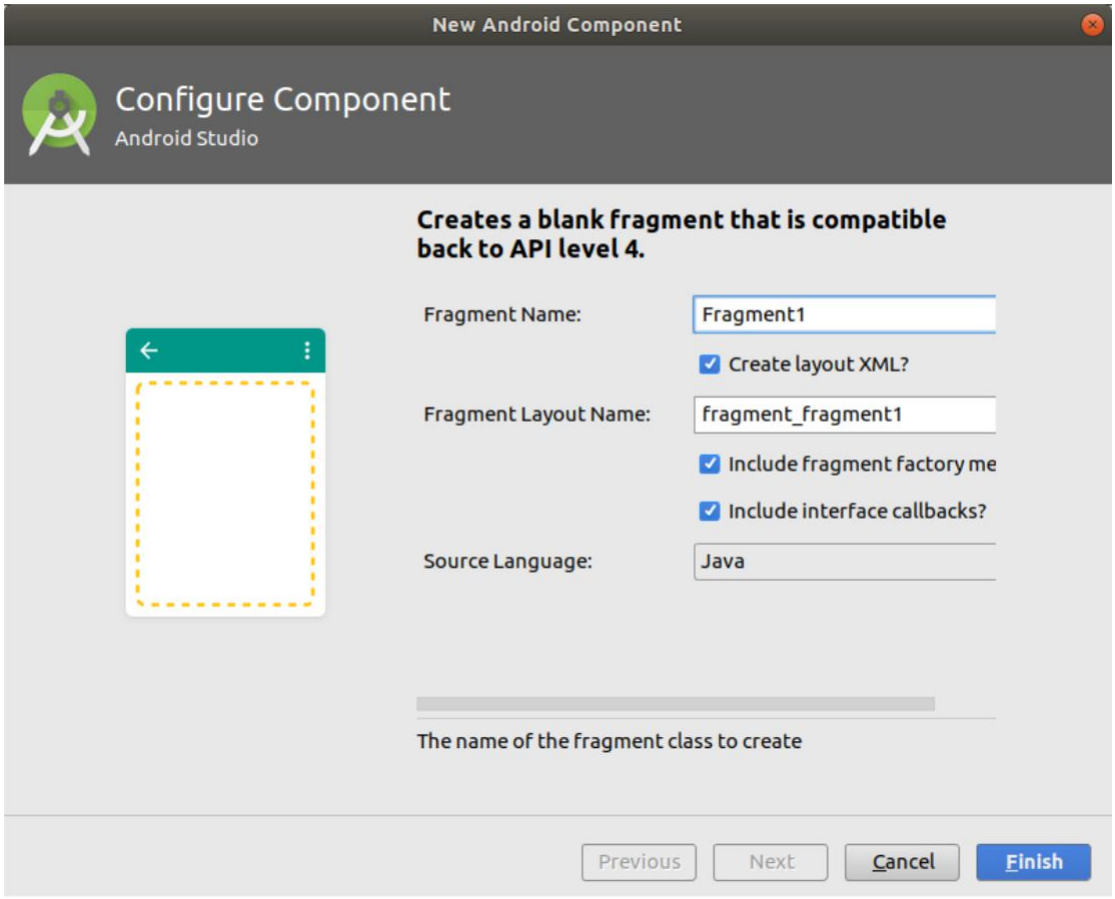
d) Right click the “LinearLayout” and convert the orientation as follows

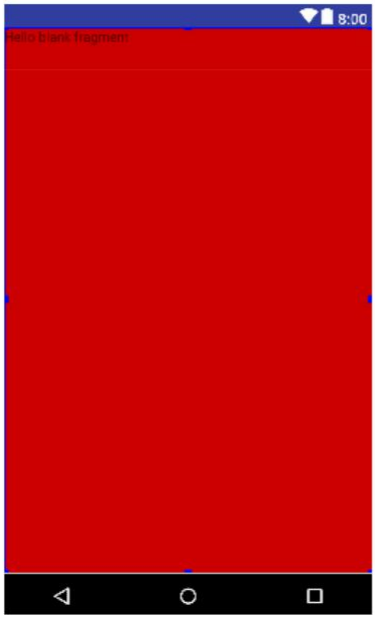


3. Right click the layout folder and go to New -> Fragment -> Fragment(Blank)



Name the Fragment “Fragment 1”





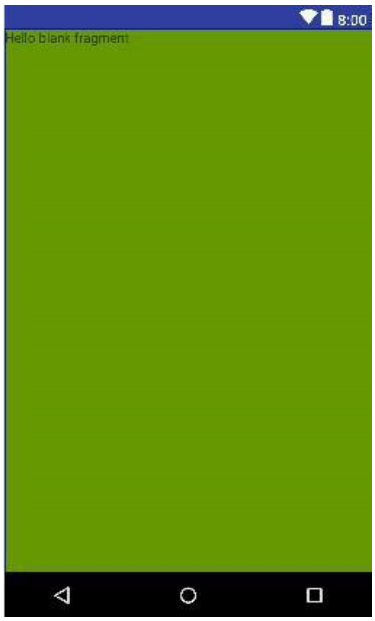
4. Repeat the above step to create a new fragment called “Fragment 2”.
5. Delete all codes within the two fragment classes keeping only the “onCreateView” method.

```
1   package com.example.thusithanjana.fragmenttest;
2
3   import ...
10
11  public class Fragment2 extends Fragment {
12
13      @Override
14      public View onCreateView(LayoutInflater inflater, ViewGroup container,
15                               Bundle savedInstanceState) {
16          // Inflate the layout for this fragment
17          return inflater.inflate(R.layout.fragment_fragment2, container,  attachToRoot: false);
18      }
19
20
21  }
22
```

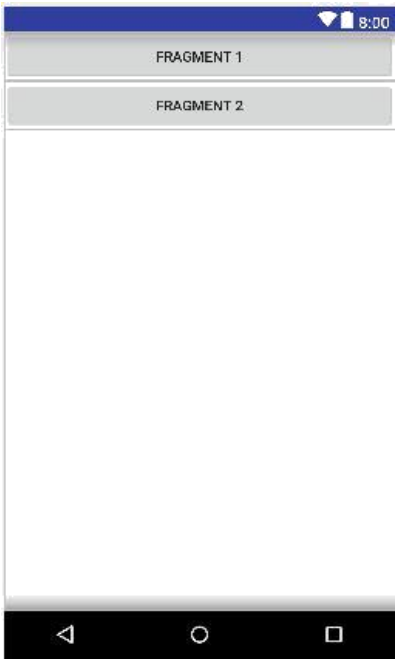
6. Go to “fragment_fragment1.xml” which is the layout file for the first fragment. Select the design surface as “Design”.

7. From attributes menu change the background color to red of the fragment

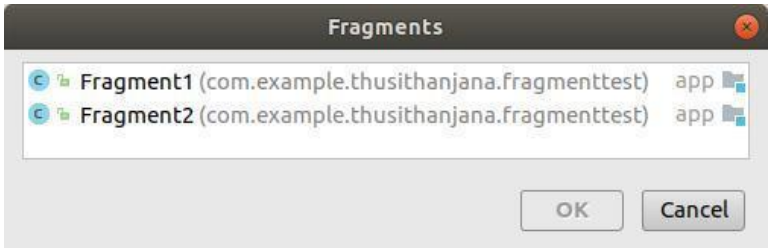
8. Follow the same step to change the color of the next fragment to green(“fragment_fragment2.xml”).



9. Go to “activity_main.xml” and design like this.



10. Add a fragment view below the Fragment 2 button. Fragment view can be found from the palette. When you drop the fragment View, you will be ask to select the default fragment. Select the Fragment1.



11. Change the layout height to “match_parent”. rename id to “fgmntDefault”
12. Designing part for this tutorial is finish. Now go to main activity.

13. Implement a method as follows.

```
public void changeFragment(View view){
    Fragment fragment;

    if (view == findViewById(R.id.btnFragment1)){
        fragment = new Fragment1();
        FragmentManager fm = getSupportFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        ft.replace(R.id.frgmntDefault,fragment);
        ft.commit();
    }

    if (view == findViewById(R.id.btnFragmetn2)){
        fragment = new Fragment2();
        FragmentManager fm = getSupportFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        ft.replace(R.id.frgmntDefault,fragment);
        ft.commit();
    }

}
```

14. Finally go to “activity_main.xml” and set the onClick method to “changeFragment” on both buttons.