Menu & Fragment

Asst. Prof. Monlica Wattana, Ph.D
Department of Computer Science,
Khon Kaen University

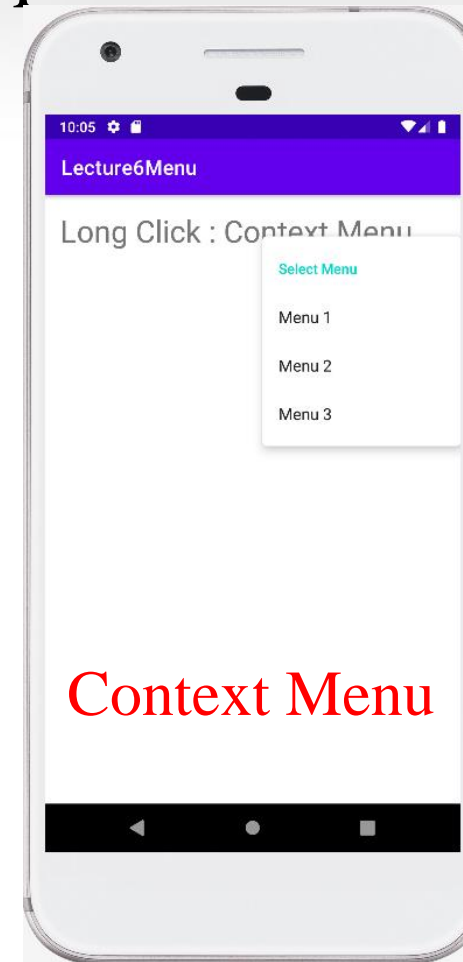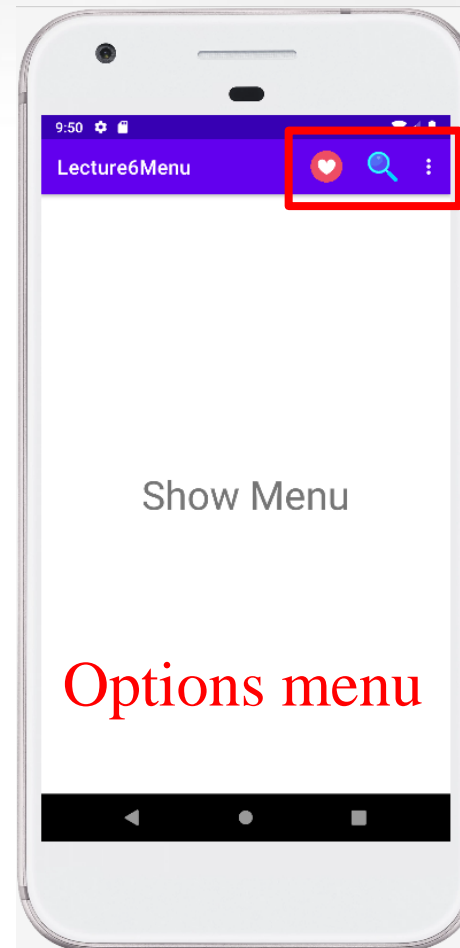342 267 Mobile Device Programming

Outline

➢ Menu

➢ Fragment

Menu

A Menu is a common user interface component in many types of applications.

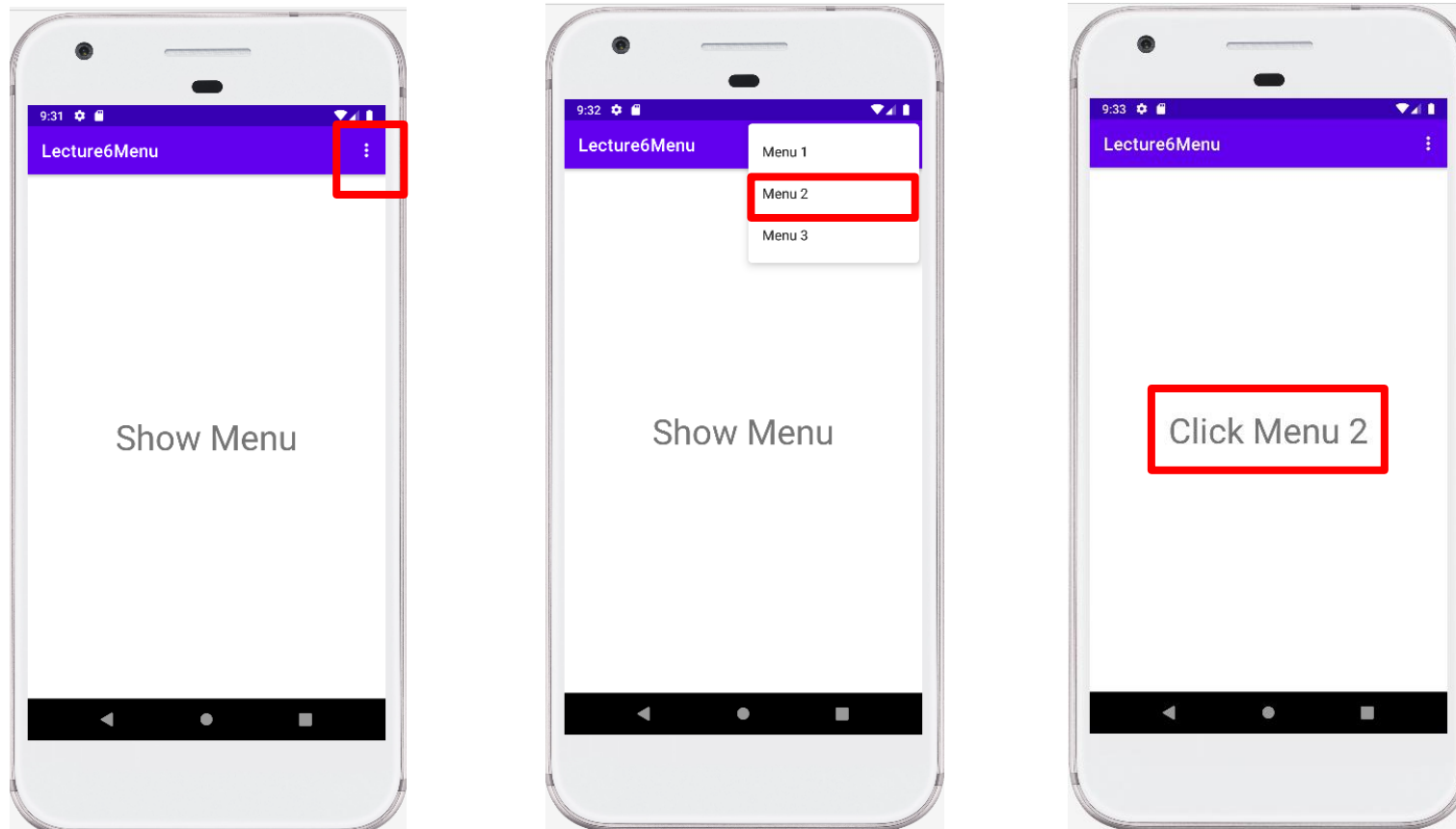**Types of Menu**
- Options menu
- Context Menu
- Popup menu

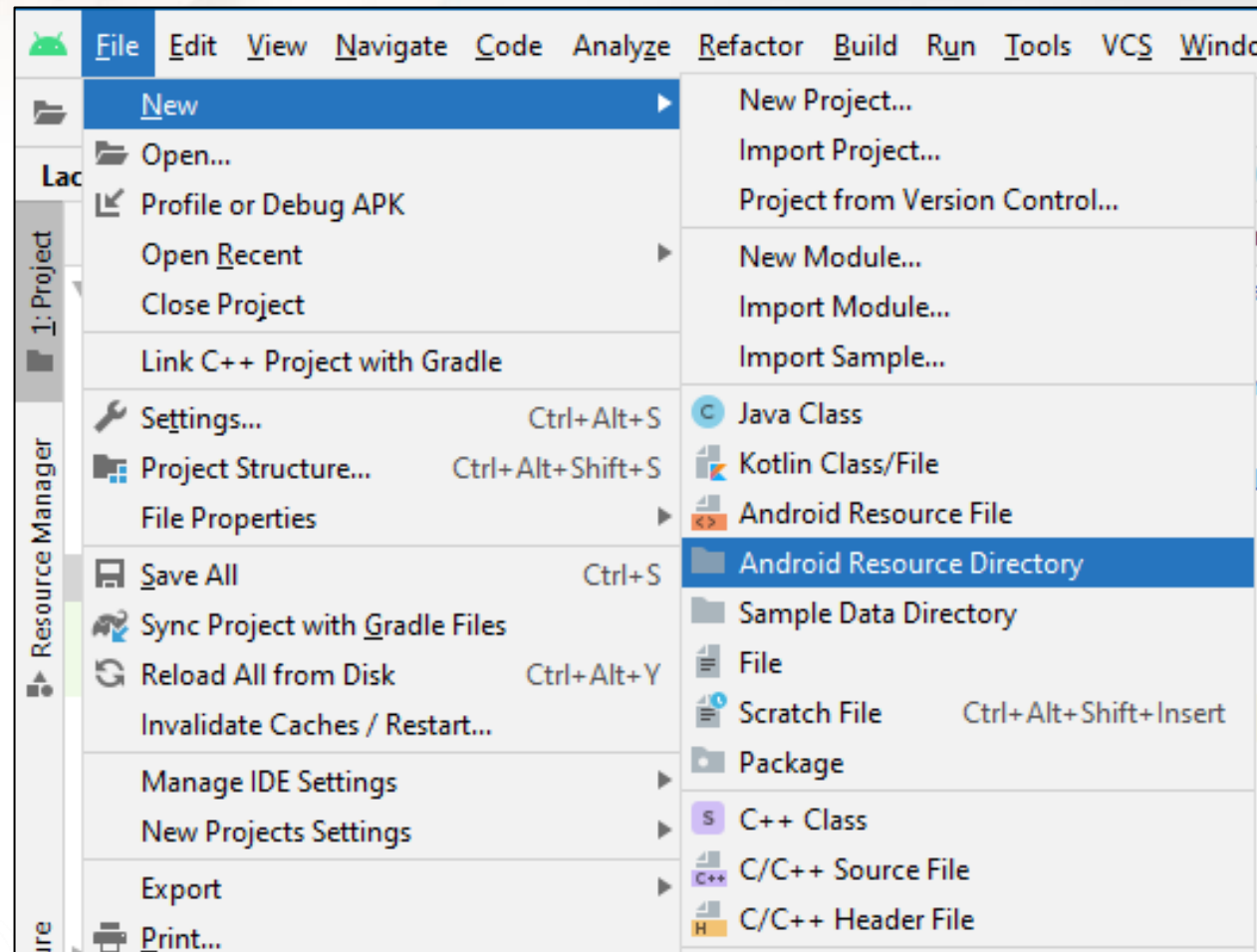Options menu

Context Menu

Popup menu

Menu: Options menu

- **Options menu:** The options menu is the primary collection of menu items for an activity.

- E.g. : 'Search', 'Compose email', and 'Settings'.

Menu: Options menu

- Create Options menu : New>> Android Resource File

Menu: Options menu

- Create Options menu (cont.)
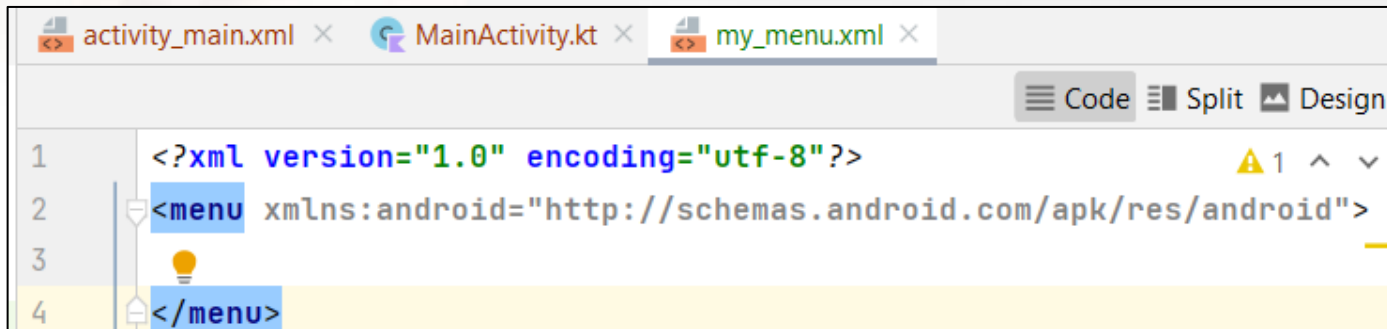
Menu: Options menu
- Create Options menu (cont.)
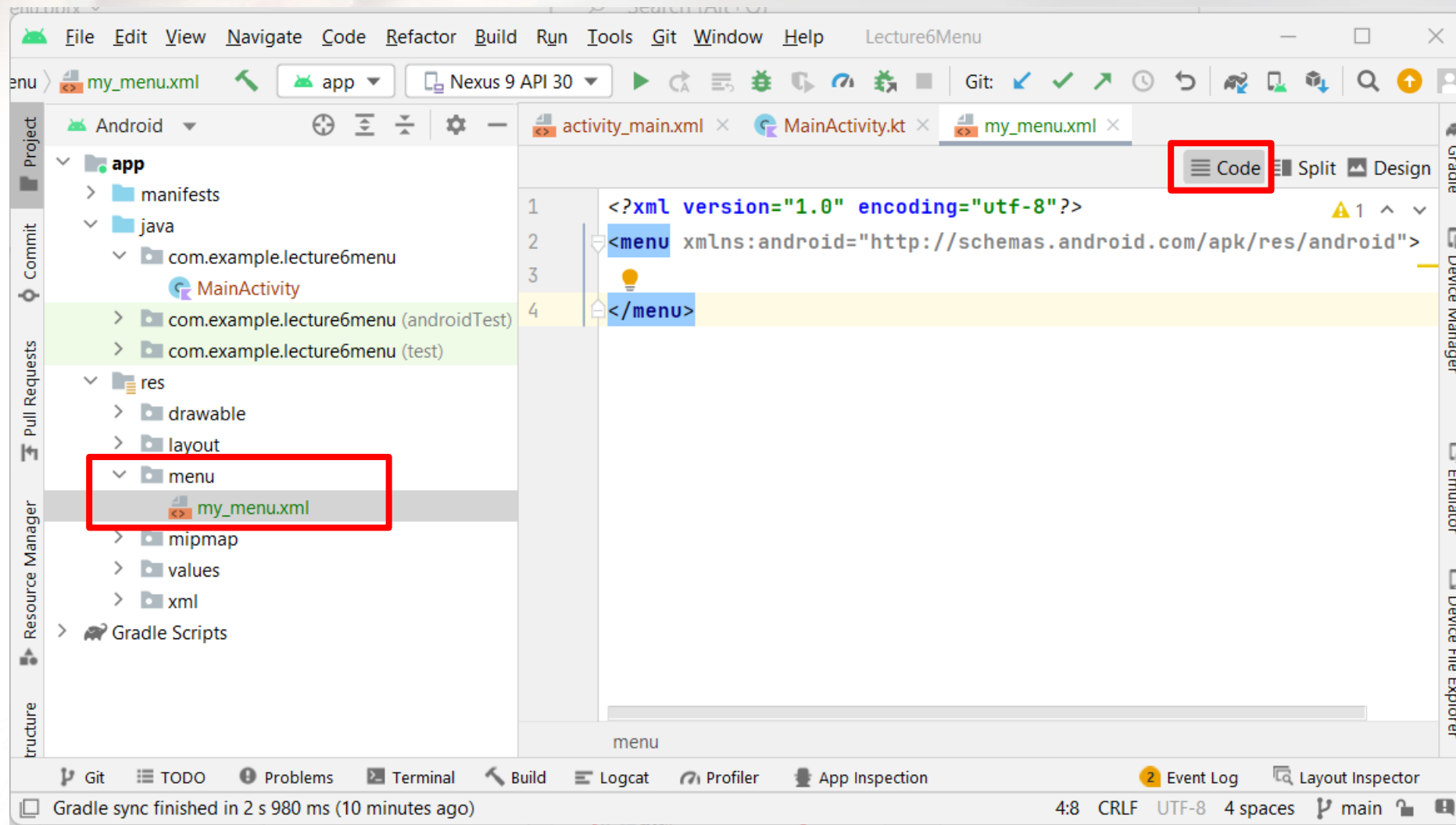
Menu: Options menu

- res>> menu>> my_menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

</menu>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item  android:id="@+id/menu1"
           android:title="Menu 1"/>
    <item  android:id="@+id/menu2"
           android:title="Menu 2"/>
    <item  android:id="@+id/menu3"
           android:title="Menu 3" />
</menu>
```
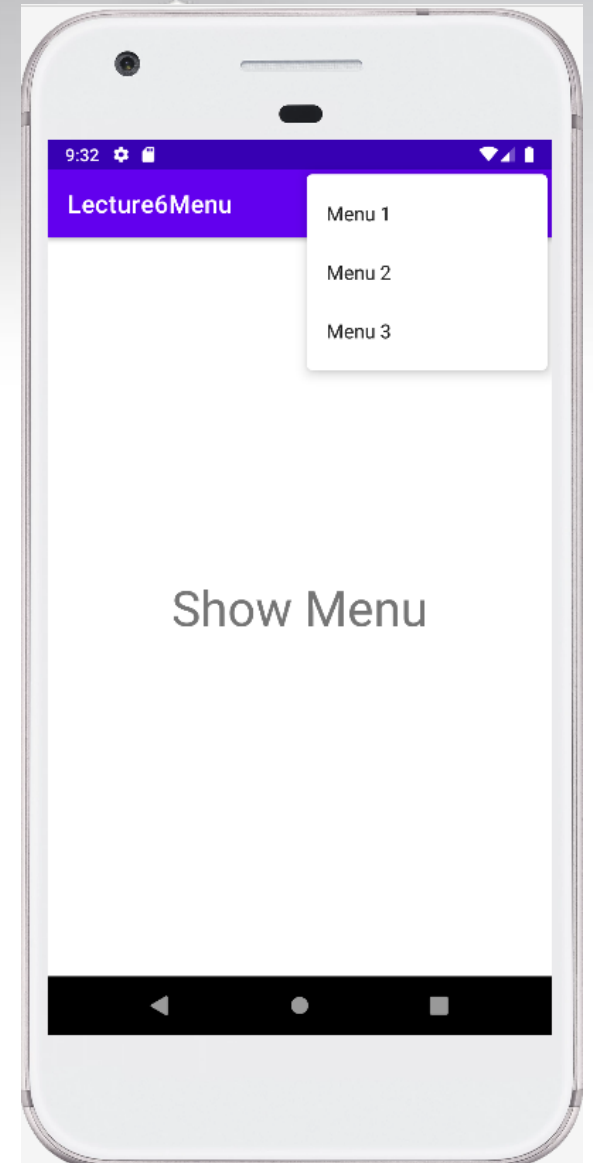
Menu: Options menu

- activity_main.xml

```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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Menu"
        android:layout_centerInParent="true"
        android:textSize="40sp"/>
</RelativeLayout>
```

Menu: Options menu

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.my_menu, menu)
        return super.onCreateOptionsMenu(menu)

    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.menu1 -> binding.textView.text = "Click Menu 1"
            R.id.menu2 -> binding.textView.text = "Click Menu 2"
            R.id.menu3 -> binding.textView.text = "Click Menu 3"
            else -> binding.textView.text = "No Selected"
        }
        return super.onOptionsItemSelected(item)
}}
```

Create menu

Menu: Option Menu with Icon

Option Menu with Icon
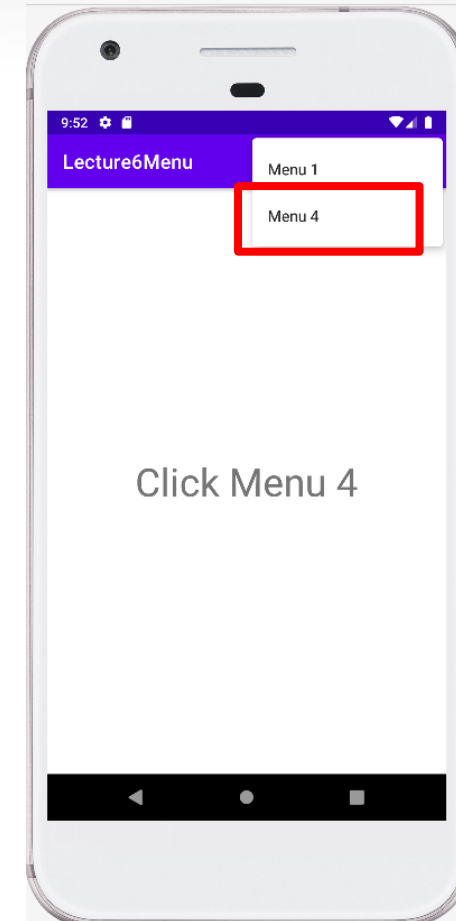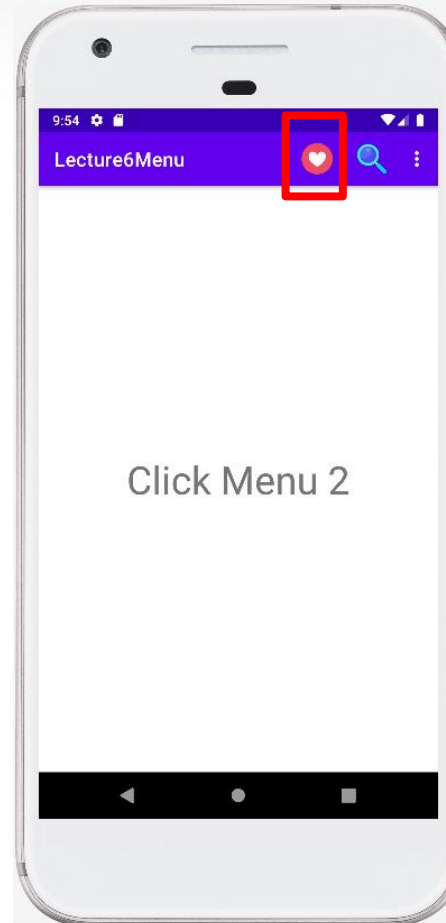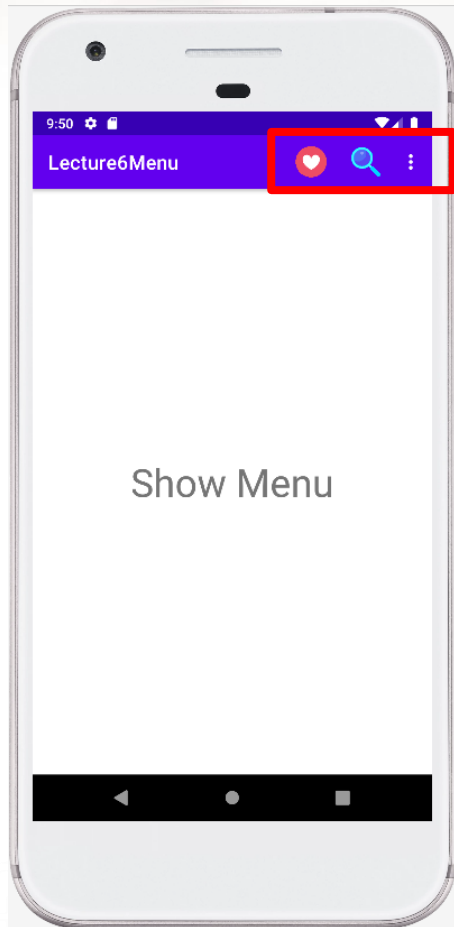
Menu: Option Menu with Icon

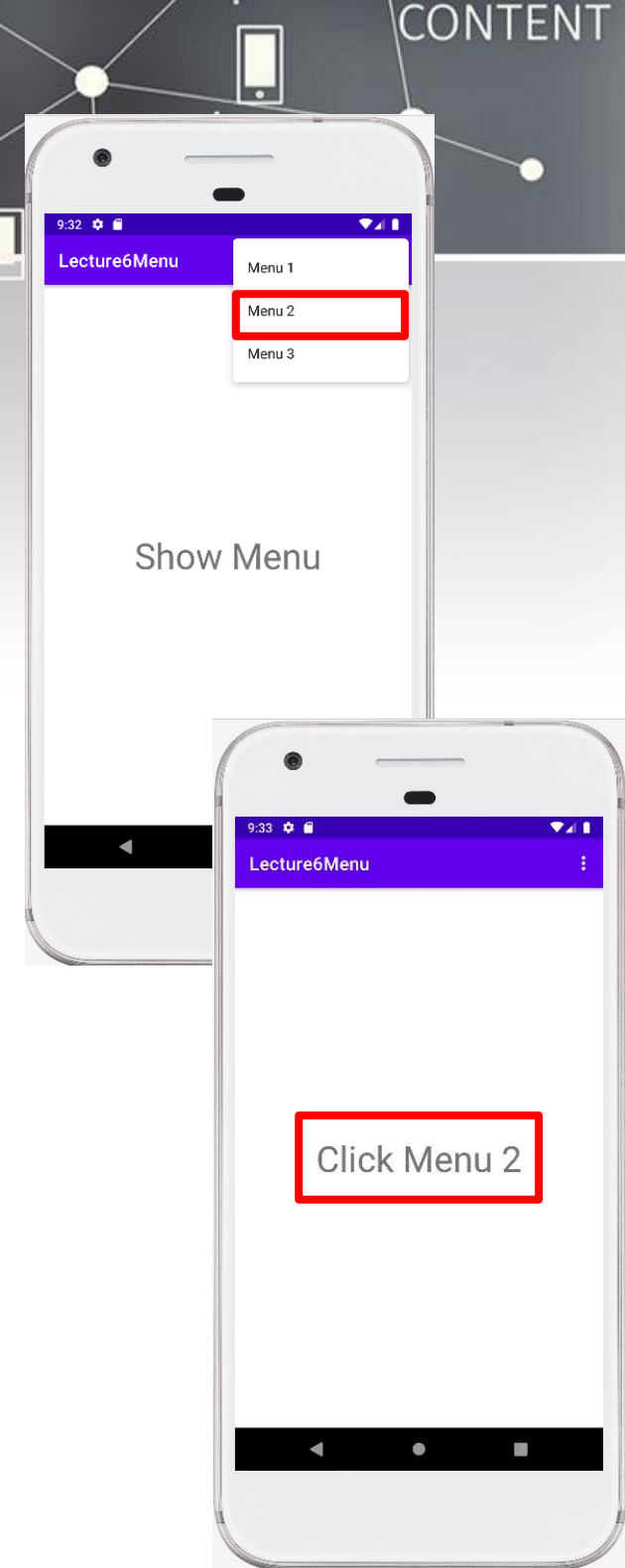## - res>> menu>> my_menu_icon.xml
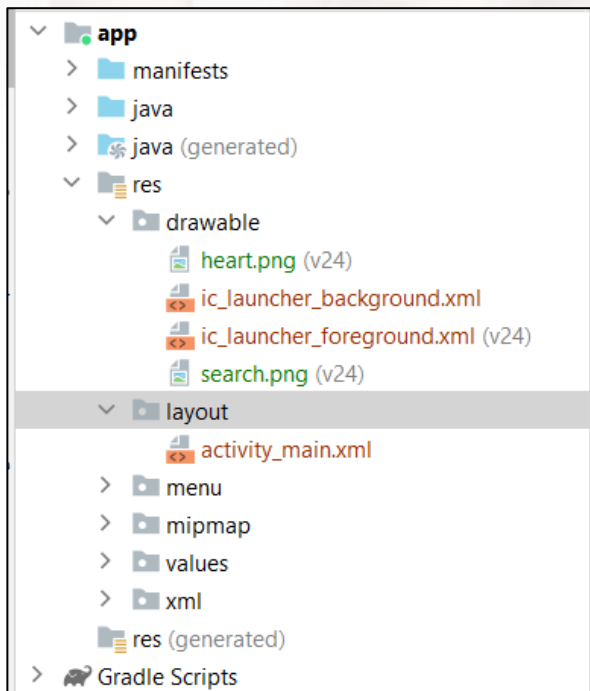
```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item  android:id="@+id/menu1"
        android:title="Menu 1"/>
    <item  android:id="@+id/menu2"
        android:title="Menu 2"
        android:icon="@drawable/heart"
        app:showAsAction="always"/>
    <item  android:id="@+id/menu3"
        android:title="Menu 3"
        android:icon="@drawable/search"
        app:showAsAction="ifRoom"/>
    <item  android:id="@+id/menu4"
        android:title="Menu 4"/>
</menu>
```
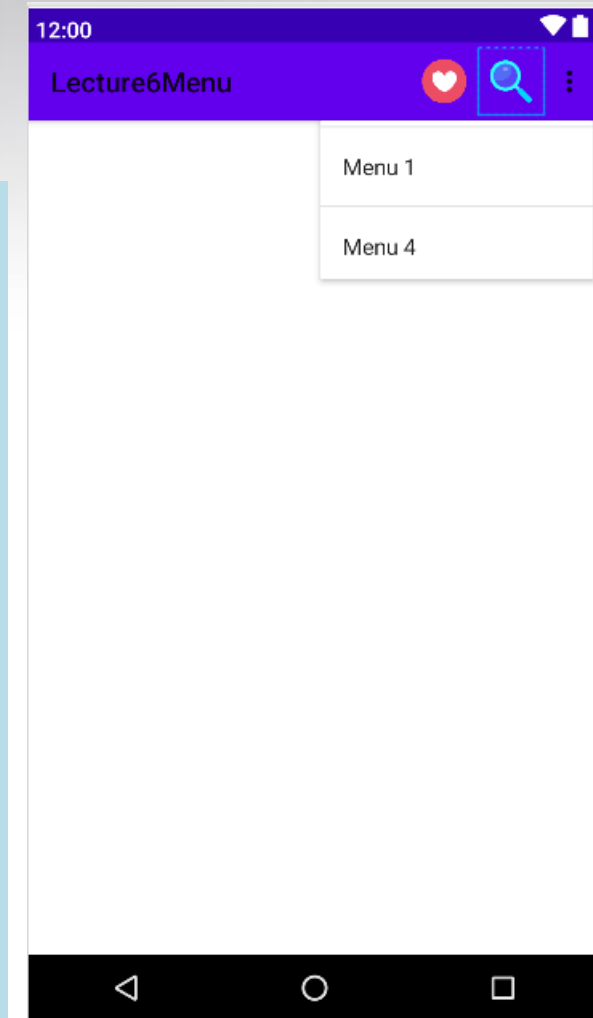
Menu: Options menu with Icon

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.my_menu_icon, menu)
        return super.onCreateOptionsMenu(menu)
    }
    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.menu1 -> binding.textView.text = "Click Menu 1"
            R.id.menu2 -> binding.textView.text = "Click Menu 2"
            R.id.menu3 -> binding.textView.text = "Click Menu 3"
            R.id.menu4 -> binding.textView.text = "Click Menu 4"
            else -> binding.textView.text = "No Selected"
        }
        return super.onOptionsItemSelected(item)
    }}
```

13
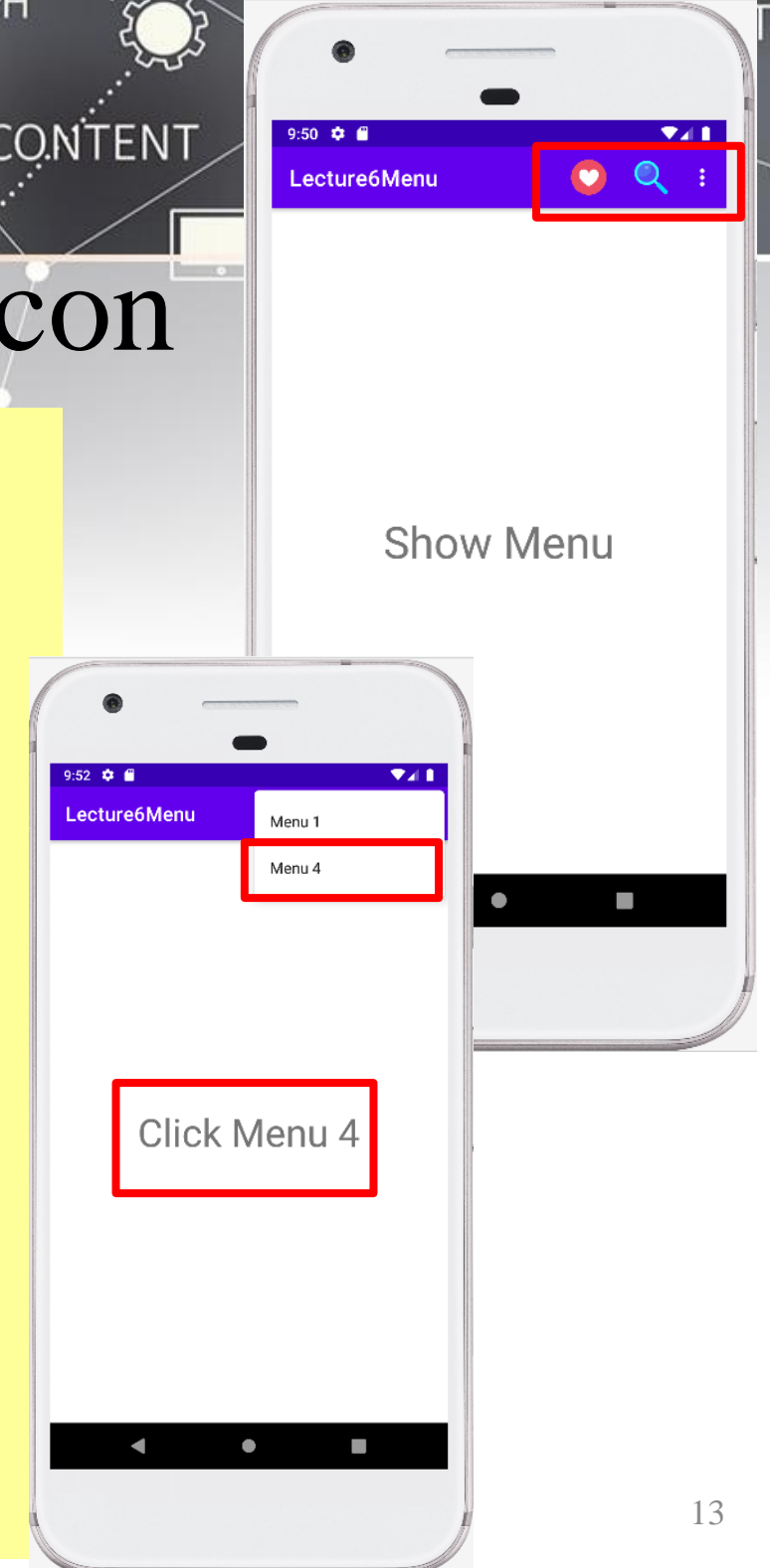
Menu: Context Menus

- Context Menu are "right-click" Menu
- Android supports context menus through an action called a *long click*
- A long click is a mouse click held down slightly longer than usual on any Android view
- Context Menus are associated with views

Using Context Menus

- Implementing a context menu:
  - Register a view for a context menu
  - Populate the context menu
  - Respond to context-menu clicks.

Register a view for a context menu

- View registration for the context menu is done in onCreate() method of an Activity

- The registration is done using registerForContextMenu method Ex.

registerForContextMenu(this.getTextView());

Populating a Context Menu

- Once a view is registered for context menus, Android will call the onCreateContextMenu() method with this view as the argument

- This is where you can populate the context menu items for that context menu (same as regular menu item)

```kotlin
override fun onCreateContextMenu(menu: ContextMenu?,
v: View?, menuInfo: ContextMenu.ContextMenuInfo?) {

    super.onCreateContextMenu(menu, v, menuInfo)
        …
}
```

Responding to Context Menu Items

- Android provides a callback method called onContextItemSelected()

Menu: Context Menu

Menu: Context Menu

- res>> menu>> my_menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item  android:id="@+id/menu1"
        android:title="Menu 1"/>
    <item  android:id="@+id/menu2"
        android:title="Menu 2"/>
    <item  android:id="@+id/menu3"
        android:title="Menu 3" />
</menu>
```
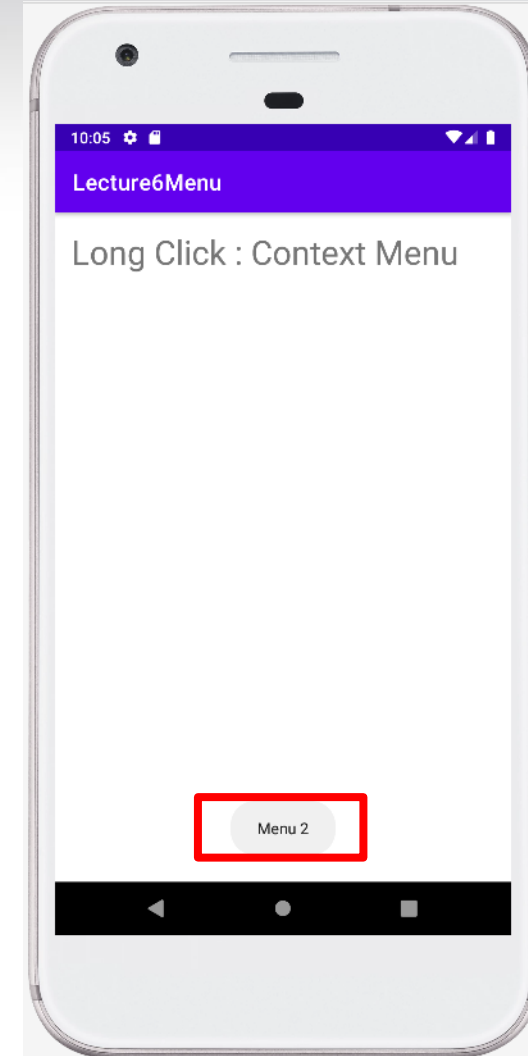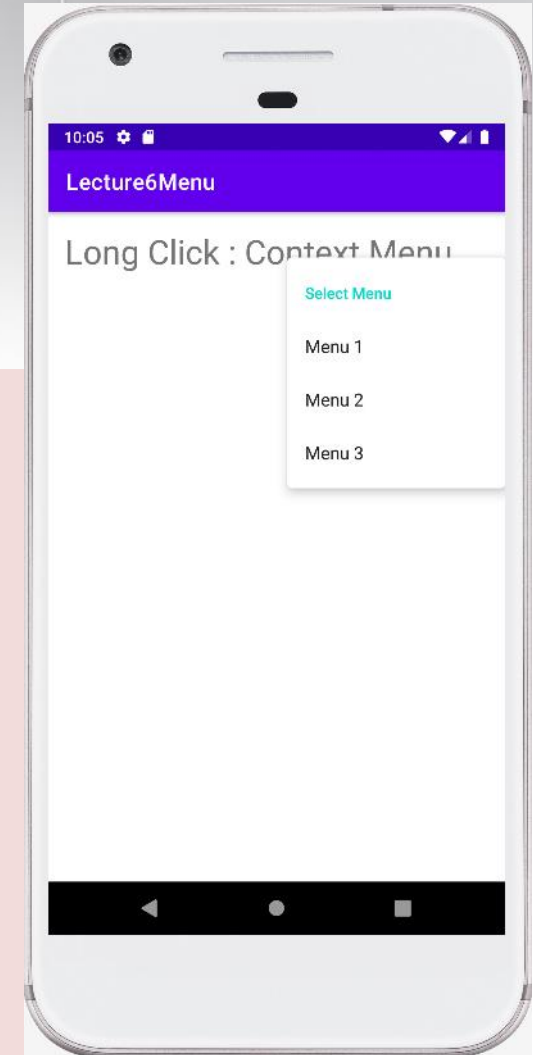
Menu: Context Menu

- res>> layout>>activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    android:padding="15dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Long Click : Context Menu"
        android:textSize="30sp"/>
</LinearLayout>
```
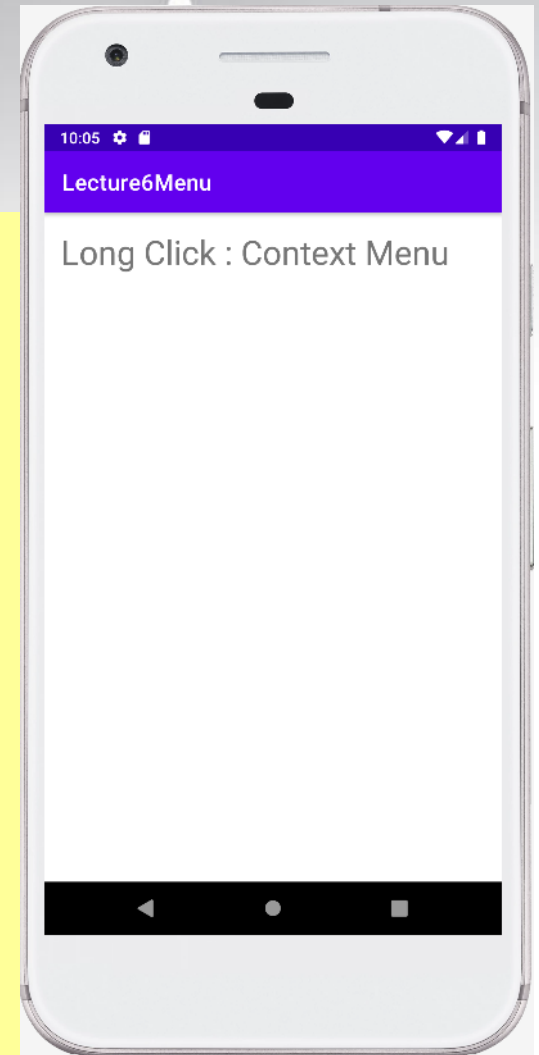
Menu: Context Menu

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {
 private lateinit var binding : ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        registerForContextMenu(binding.textView)
    }

    override fun onCreateContextMenu(menu: ContextMenu?, v: View?, menuInfo: ContextMenu.ContextMenuInfo?)
{
        super.onCreateContextMenu(menu, v, menuInfo)
        menuInflater.inflate(R.menu.my_menu, menu)
        menu?.setHeaderTitle("Select Menu")
    }
```
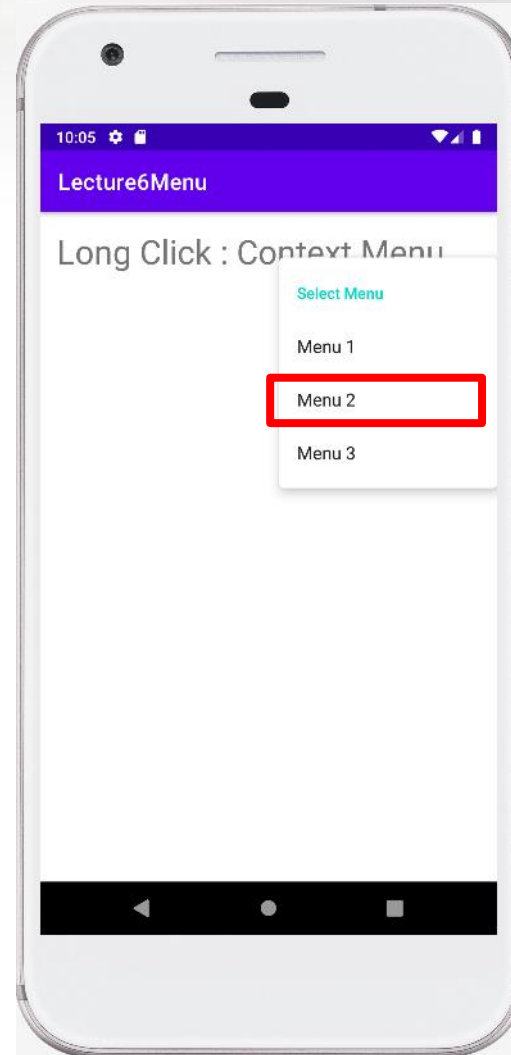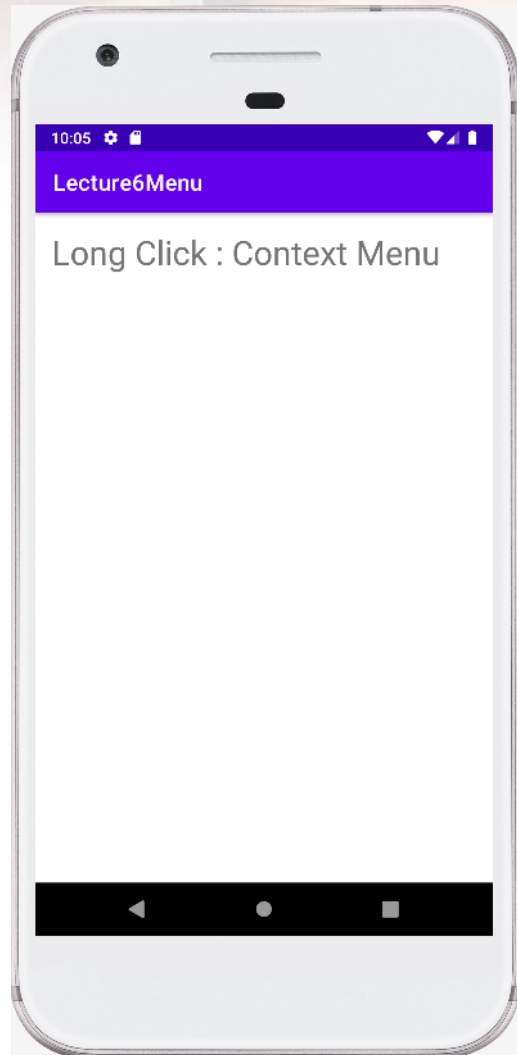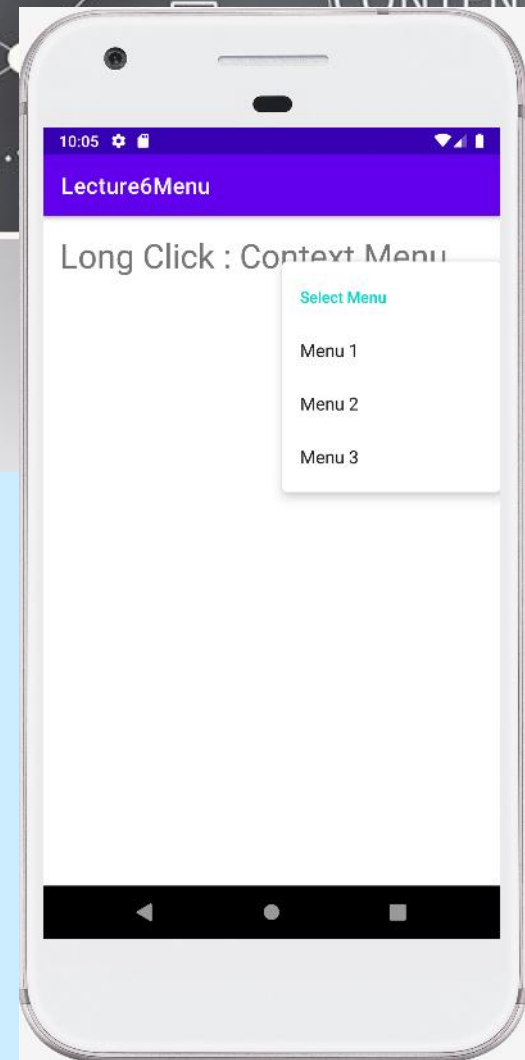
Menu: Context Menu

MainActivity.kt (cont)

```kotlin
override fun onContextItemSelected(item: MenuItem): Boolean {
    when (item?.itemId) {
        R.id.menu1 -> {
            Toast.makeText(applicationContext, item.title, Toast.LENGTH_LONG).show()
            return true        }
        R.id.menu2 ->{
            Toast.makeText(applicationContext, item.title, Toast.LENGTH_LONG).show()
            return true        }
        R.id.menu3 ->{
            Toast.makeText(applicationContext, item.title, Toast.LENGTH_LONG).show()
            return true        }
        else -> return super.onOptionsItemSelected(item)
    }
}
```
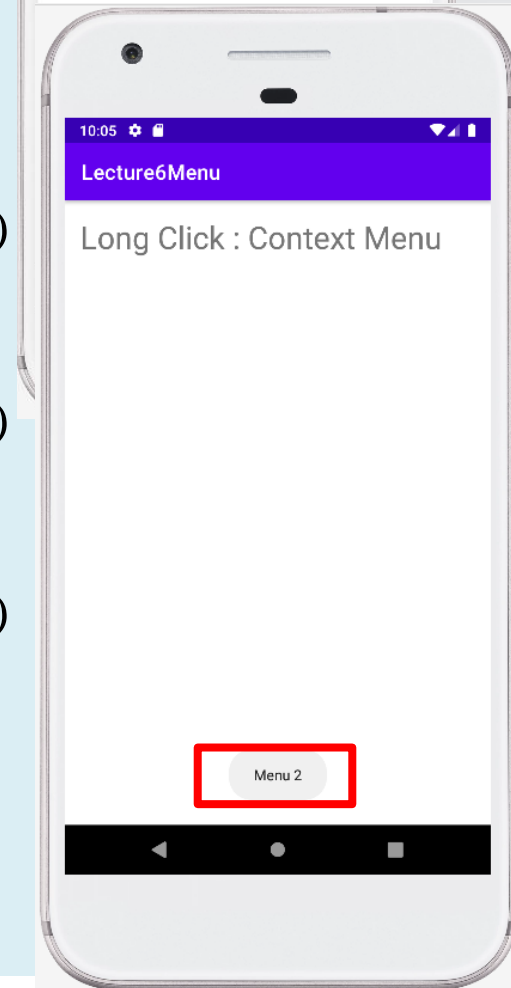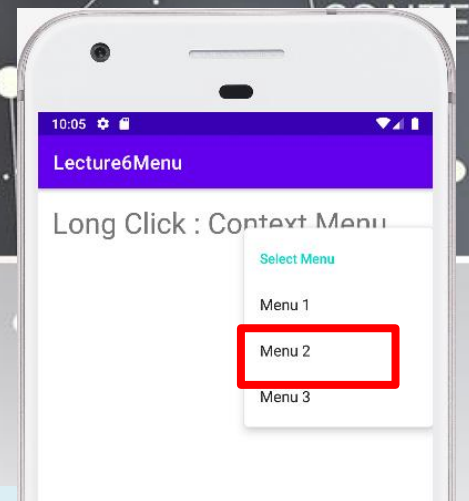
Menu: Popup menu

- Menu displays the menu below the anchor text if space is available otherwise above the anchor text.
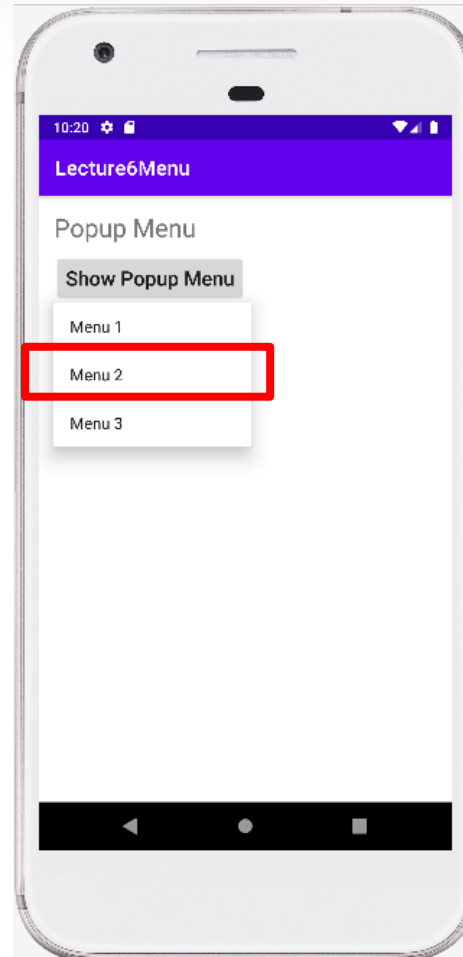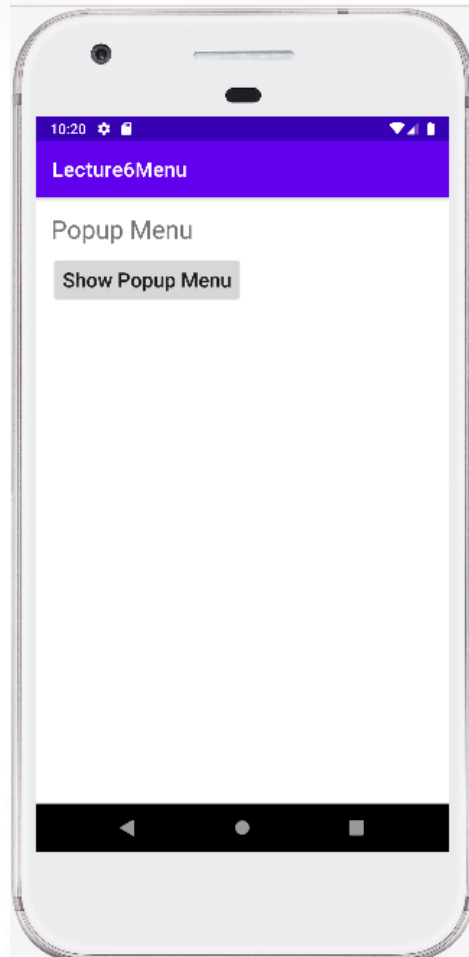
Menu: Popup menu

- res>> menu>> my_menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item  android:id="@+id/menu1"
        android:title="Menu 1"/>
    <item  android:id="@+id/menu2"
        android:title="Menu 2"/>
    <item  android:id="@+id/menu3"
        android:title="Menu 3" />
</menu>
```
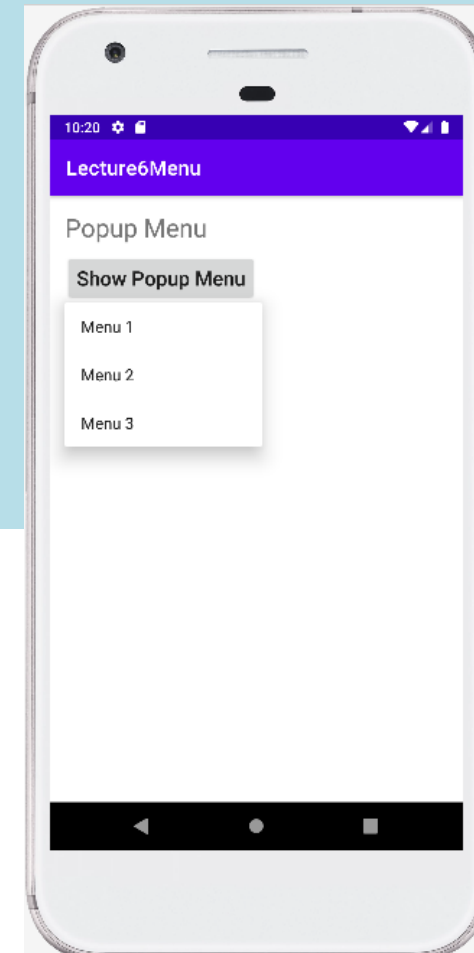
Menu: Popup menu

res>> layout
>>activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">
   <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Popup Menu "
    android:textSize="25sp"/>
 <androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btn_popMenu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Popup Menu"
    android:textSize="20sp"
    android:textAllCaps="false"
    android:layout_marginTop="10dp"
    android:onClick="showPopupMenu"/></LinearLayout>
```
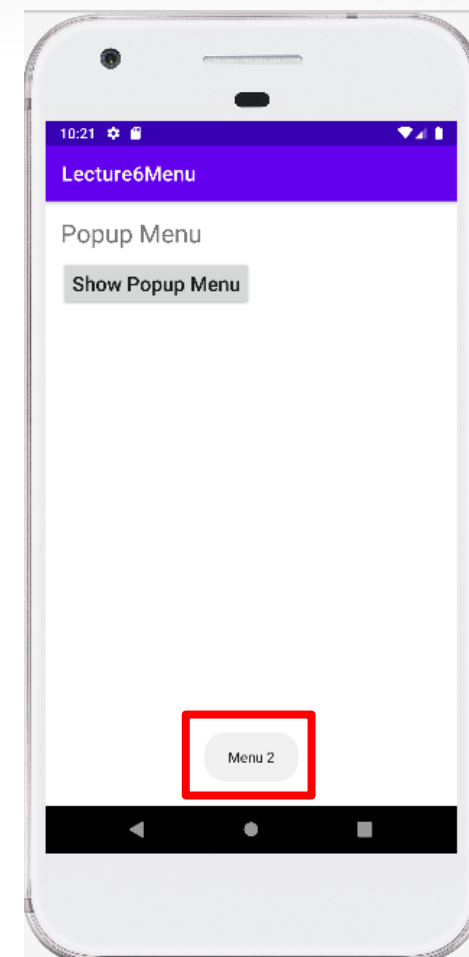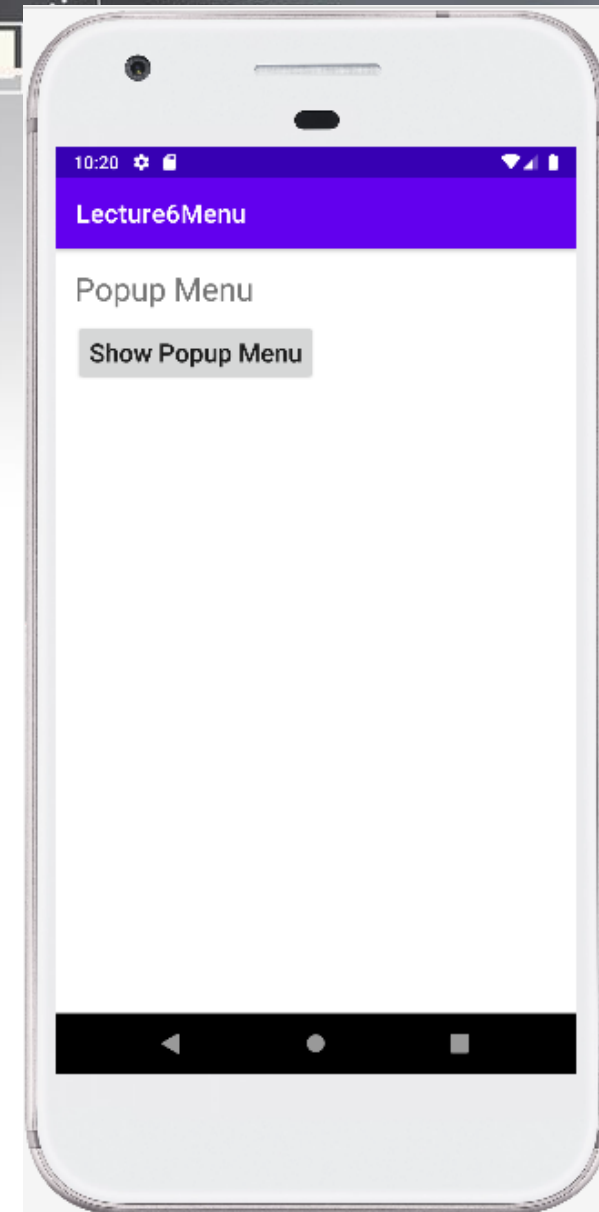
Menu:
Popup menu

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }

    fun showPopupMenu(v:View){
        //Creating the instance of PopupMenu
        val popup = PopupMenu(this, binding.btnPopMenu)
        //Inflating the Popup using xml file
        popup.inflate(R.menu.my_menu)

        popup.setOnMenuItemClickListener(PopupMenu.OnMenuItemClickListener {
            item: MenuItem? ->
            when (item!!.itemId) {
                R.id.menu1 -> Toast.makeText(this, item.title, Toast.LENGTH_SHORT).show()
                R.id.menu2 -> Toast.makeText(this, item.title, Toast.LENGTH_SHORT).show()
                R.id.menu3 -> Toast.makeText(this, item.title, Toast.LENGTH_SHORT).show() }
            true    })
        popup.show()
    }
}
```
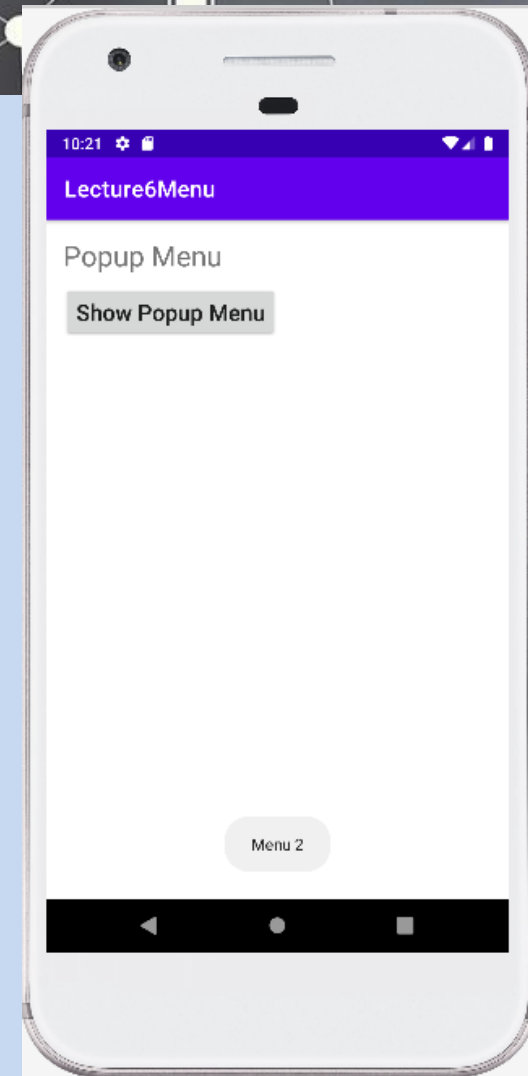
Fragment

- An activity is a container for views
- When you have a larger screen device than a phone –like a tablet it can look too simple to use phone interface here.
- Fragments
    - Mini-activities, each with its own set of views
    - One or more fragments can be embedded in an Activity
    - You can do this dynamically as a function of the device type (tablet or not) or orientation
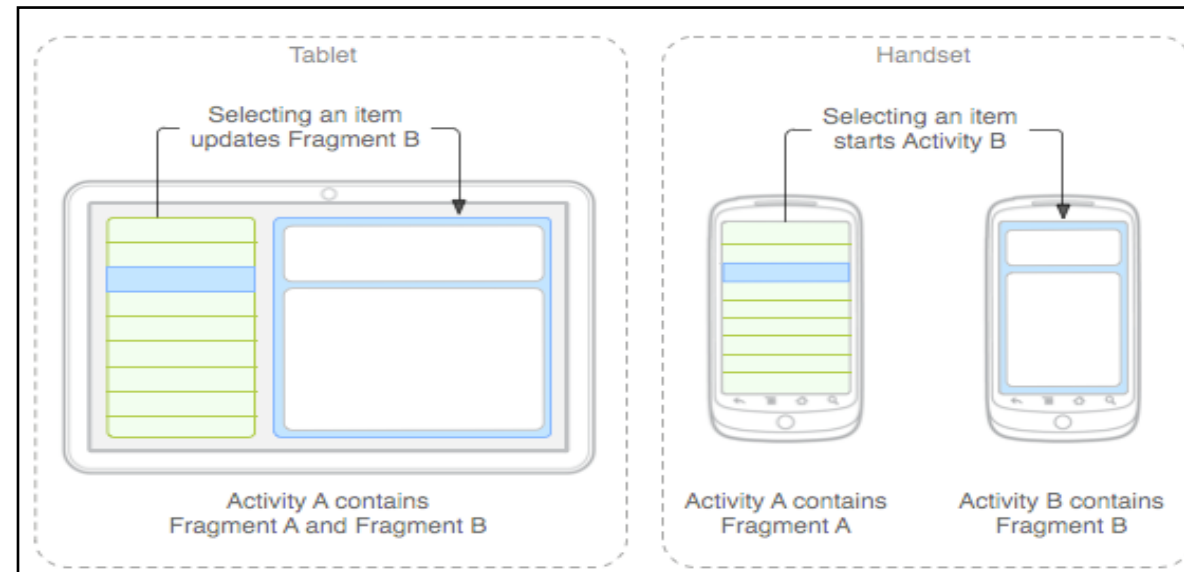
Fragment Lifecycle

- Fragment in an Activity---Activity Lifecyle influences
  - Activity paused → all its fragments paused
  - Activity destroyed → all its fragments paused
  - Activity running → manipulate each fragment independently.
- Fragment transaction → add, remove, replace, etc.
  - adds it to a back stack that's managed by the activity—each back stack entry in the activity is a record of the fragment transaction that occurred.
  - The back stack allows the user to reverse a fragment transaction (navigate backwards), by pressing the *Back button*.
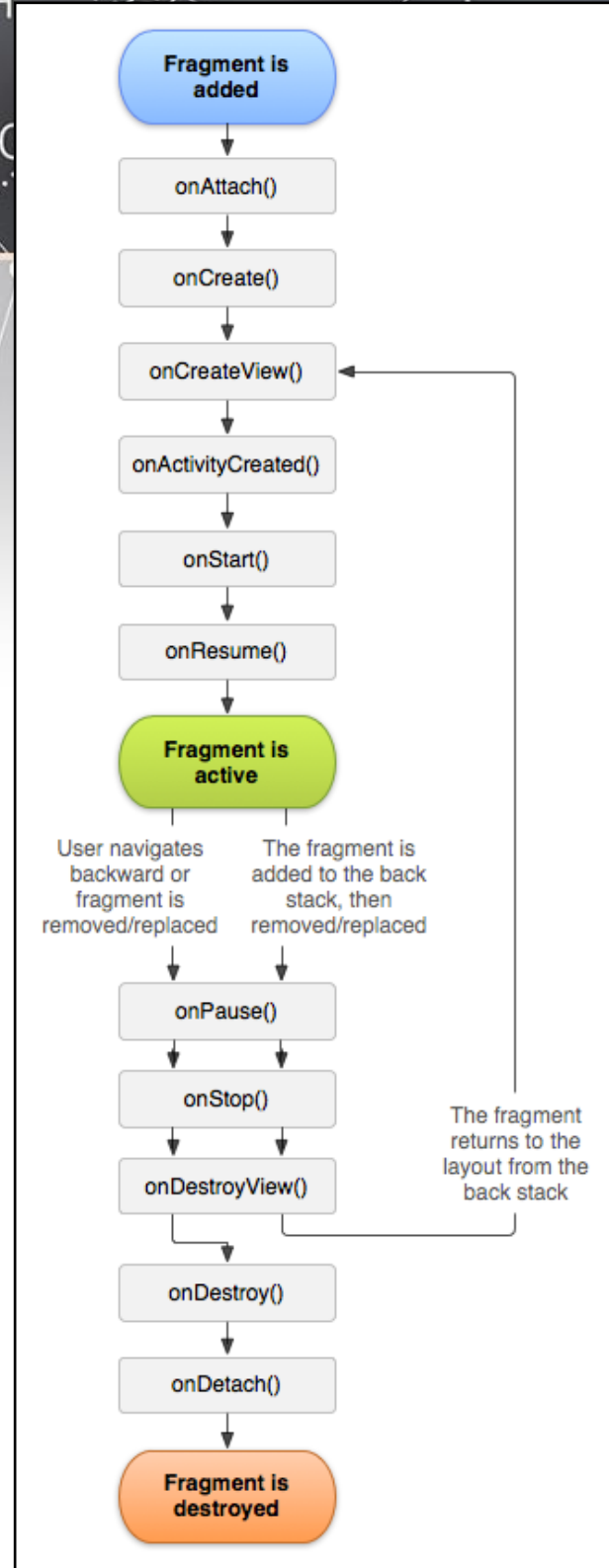
Fragment methods (callback functions)

- onAttach(Activity) called once the fragment is associated with its activity.

- onCreate(Bundle) called to do initial creation of the fragment.

- onCreateView(LayoutInflater, ViewGroup, Bundle) creates and returns the view hierarchy associated with the fragment.

- onActivityCreated(Bundle) tells the fragment that its activity has completed its own Activity.onCreate.

- onStart() makes the fragment visible to the user (based on its containing activity being started).

- onResume() makes the fragment interacting with the user (based on its containing activity being resumed).

Fragment methods (callback functions)

As a fragment is no longer being used, it goes through a reverse series of callbacks:

- onPause() fragment is no longer interacting with the user either because its activity is being paused or a fragment operation is modifying it in the activity.

- onStop() fragment is no longer visible to the user either because its activity is being stopped or a fragment operation is modifying it in the activity.

- onDestroyView() allows the fragment to clean up resources associated with its View.

- onDestroy() called to do final cleanup of the fragment's state.

- onDetach() called immediately prior to the fragment no longer being associated with its activity.

Fragments and their UI – onCreateView() using XML

- Can implement onCreateView using XML

Activity parent's ViewGroup

```
class ExampleFragment :Fragment {

    override fun onCreateView( inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?  ): View? {
```

Bundle that provides data about the previous
instance of the fragment, if the fragment is being resumed

```
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.example_fragment, container, false);
    }

}
```

example_fragment.xml file that contains the layout
This will be contained in resource layout folder.

Adding Fragment

There are 2 options for adding fragment.

1. Adding to an Activity via layout XML.

2. Creating and adding to an Activity via CODE.

OPTION 1: Adding to an Activity via layout XML.

OPTION 1: Adding to an Activity via layout XML

OPTION 1:
Adding to an Activity
via layout XML.

Tabs: FirstFragment.kt | activity_main.xml | MainActivity.kt | fragment_first.xml

```kotlin
package com.example.lecture6menu

import ...

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [FirstFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class FirstFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
```

Tabs: fragment_first.xml | FirstFragment.kt | activity_main.xml | MainActivity.kt | my_menu_

```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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=".FirstFragment">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="Hello blank fragment" />

</FrameLayout>
```

Project tree:
- app
  - manifests
  - java
    - com.example.lecture6menu
      - FirstFragment.kt
      - MainActivity
    - com.example.lecture6menu (androidTest)
    - com.example.lecture6menu (test)
  - java (generated)
  - res
    - drawable
    - layout
      - activity_main.xml
      - fragment_first.xml
    - menu
    - mipmap
    - values
    - xml
  - res (generated)
- Gradle Scripts

OPTION 1 –adding to an Activity via layout XML.

OPTION 1: Adding to an Activity via layout XML.

fragment_first.xml

fragment_second.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:background="#A7FFEB"
    android:padding="20dp"
    tools:context=".FirstFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickFirstFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Click Me" /></LinearLayout>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:background="#F8B6B1"
    android:padding="20dp"
    tools:context=".SecondFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickSecondFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Click Me" /></LinearLayout>
```

OPTION 1: Adding to an Activity via layout XML.

FirstFragment.kt


```kotlin
package com.example.lecture6menu

import ...

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [FirstFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class FirstFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {  it: Bundle
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
```

```kotlin
package com.example.lecture6menu

import ...

class FirstFragment : Fragment() {
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_first, container,  attachToRoot: false)
    }
}
```

OPTION 1: Adding to an Activity via layout XML.

- FirstFragment.kt

```kotlin
class FirstFragment : Fragment() {
    private lateinit var bindingFirst : FragmentFirstBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        bindingFirst = FragmentFirstBinding.inflate(layoutInflater)
        bindingFirst.btnClickFirstFrag.setOnClickListener(){
            val toast = Toast.makeText(context, "Click On First Fragment", Toast.LENGTH_SHORT)
            toast.show()
        }
        return bindingFirst.root
    }
}
```
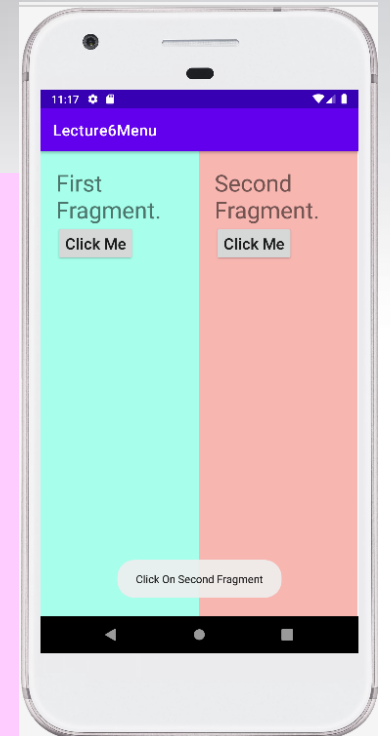
OPTION 1: Adding to an Activity via Activity layout XML.

- SecondFragment.kt

```
class SecondFragment : Fragment() {
    private lateinit var bindingSecond : FragmentSecondBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        bindingSecond = FragmentSecondBinding.inflate(layoutInflater)
        bindingSecond.btnClickSecondFrag.setOnClickListener(){
            val toast = Toast.makeText(context, "Click On Second Fragment", Toast.LENGTH_SHORT)
            toast.show()
        }
        return bindingSecond.root
    }
}
```

41

OPTION 1: Adding to an Activity via layout XML.

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">

<fragment
    android:id="@+id/first_fragment"
    android:name="com.myweb.lac6menufragment.FirstFragment"
    android:layout_width="200dp"
    android:layout_height="match_parent"/>
  <fragment
    android:id="@+id/second_fragment"
    android:name="com.myweb.lac6menufragment.SecondFragment"
    android:layout_width="200dp"
    android:layout_height="match_parent"/>
</LinearLayout>
```

2

OPTION 1: Adding to an Activity via layout XML.

- MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
```

3

OPTION 2: Creating and adding to an Activity via CODE.

- Inside Activity Code where you want to add Fragment (dynamically anywhere or in onCreate() callback)

- Get FragmentTransaction associated with this Activity

 val fragmentManager : FragmentManager = supportFragmentManager()
  val fragmentTransaction : FragmentTransaction= fragmentManager.beginTransaction()

- Create instance of Fragment

  val fragment = ExampleFragment()

- Add Fragment instance to Activity

  fragmentTransaction.add(R.id.fragment_container, fragment)

  fragmentTransaction.commit()
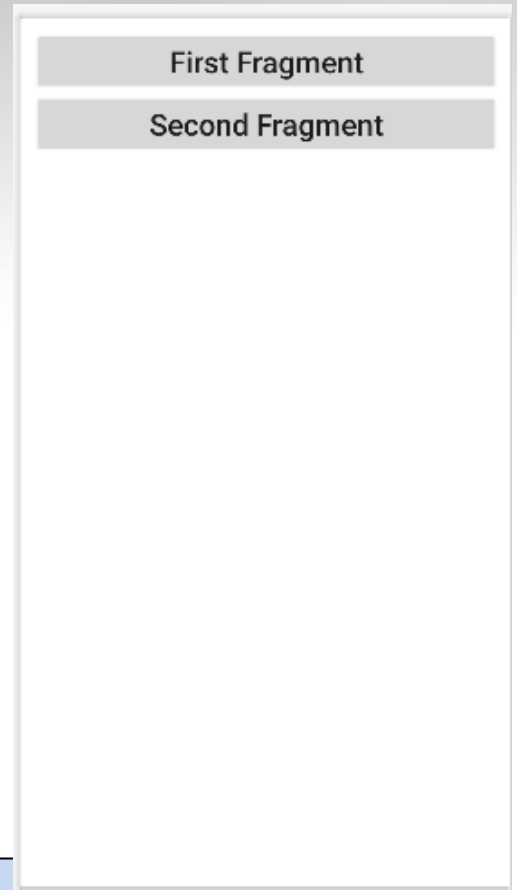
OPTION 2: Creating and adding to an Activity via CODE.
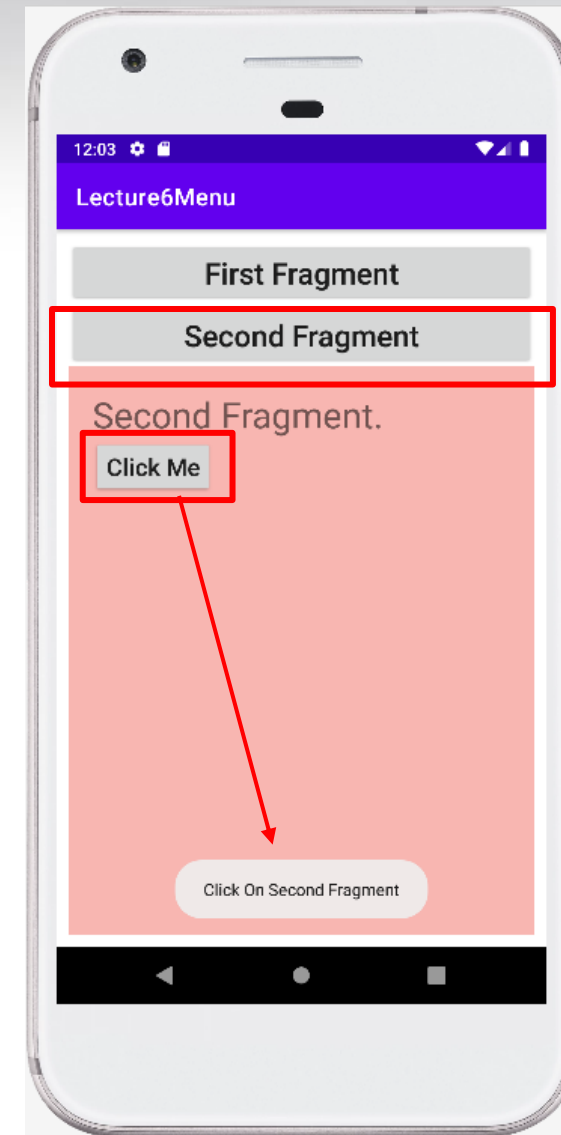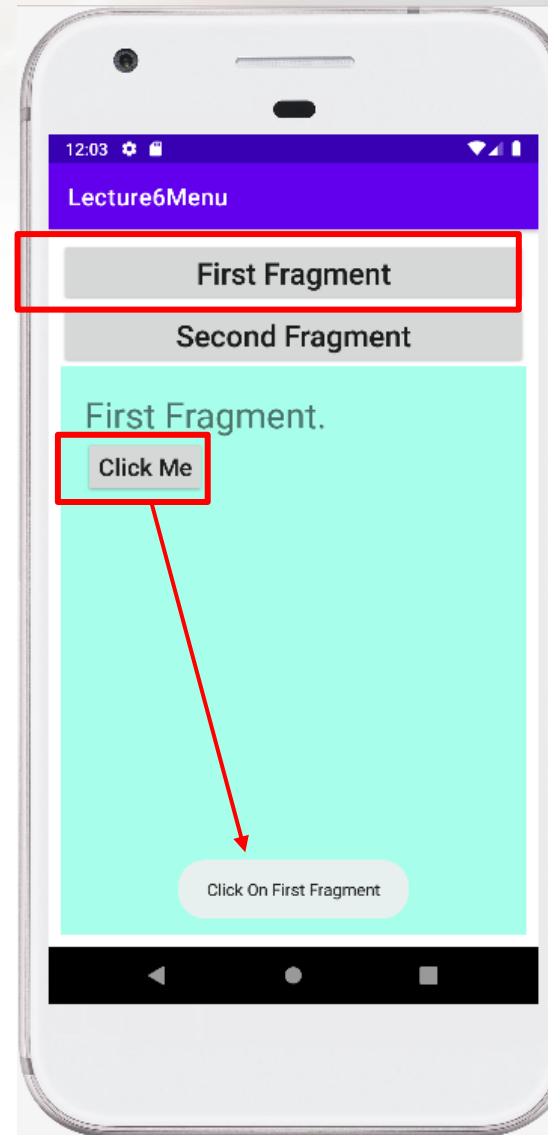
OPTION 2: Creating and adding to an Activity via CODE.

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout ….
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">
  <androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btnFirstFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="First Fragment"
    android:textSize="25sp"
    android:textAllCaps="false"
    android:onClick="clickFirstFragment"/>
  <androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btnSecondFragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Second Fragment"
    android:textSize="25sp"
    android:textAllCaps="false"
    android:onClick="clickSecondFragment"/>
```

First Fragment

Second Fragment

```xml
<FrameLayout
    android:id="@+id/frameLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
</LinearLayout>
```

OPTION 2: Creating and adding to an Activity via CODE.

fragment_first.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
    android:orientation="vertical"
    android:background="#A7FFEB"
    android:padding="20dp"
    tools:context=".FirstFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickFirstFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Click Me" /></LinearLayout>
```

First Fragment.
Click Me

fragment_second.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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"
    android:orientation="vertical"
    android:background="#F8B6B1"
    android:padding="20dp"
    tools:context=".SecondFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickSecondFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Click Me" /></LinearLayout>
```
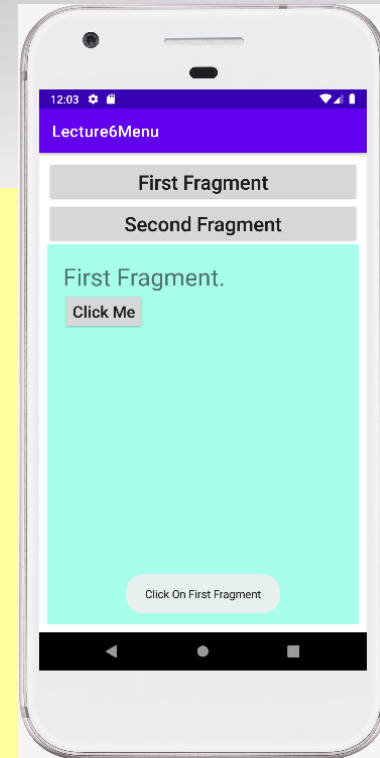
Second Fragment.
Click Me

OPTION 2: Creating and adding to an Activity via CODE.

- ## FirstFragment.kt

```kotlin
class FirstFragment : Fragment() {
    private lateinit var bindingFirst : FragmentFirstBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        bindingFirst = FragmentFirstBinding.inflate(layoutInflater)
        bindingFirst.btnClickFirstFrag.setOnClickListener(){
            val toast = Toast.makeText(context, "Click On First Fragment", Toast.LENGTH_SHORT)
            toast.show()
        }
        return bindingFirst.root
    }
}
```
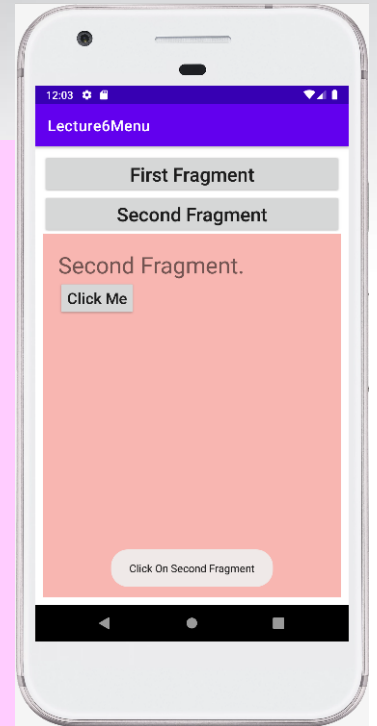
48

OPTION 2: Creating and adding to an Activity via CODE.
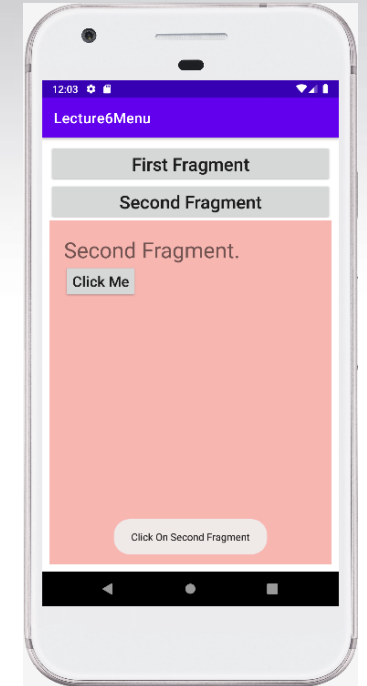
- SecondFragment.kt

```
class SecondFragment : Fragment() {
    private lateinit var bindingSecond : FragmentSecondBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        bindingSecond = FragmentSecondBinding.inflate(layoutInflater)
        bindingSecond.btnClickSecondFrag.setOnClickListener(){
            val toast = Toast.makeText(context, "Click On Second Fragment", Toast.LENGTH_SHORT)
            toast.show()
        }
        return bindingSecond.root
    }
}
```

49

# OPTION 2: Creating and adding to an Activity via CODE.

- MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {
    private lateinit var binding : ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
fun clickFirstFragment(v:View){
    supportFragmentManager.beginTransaction().add(
        R.id.frameLayout,
        FirstFragment()
    ).commit()
}
```

```kotlin
fun clickSecondFragment(v:View){
    supportFragmentManager.beginTransaction().add(
        R.id.frameLayout,
        SecondFragment()
    ).commit()
}
```

Call Another Fragment on Fragment
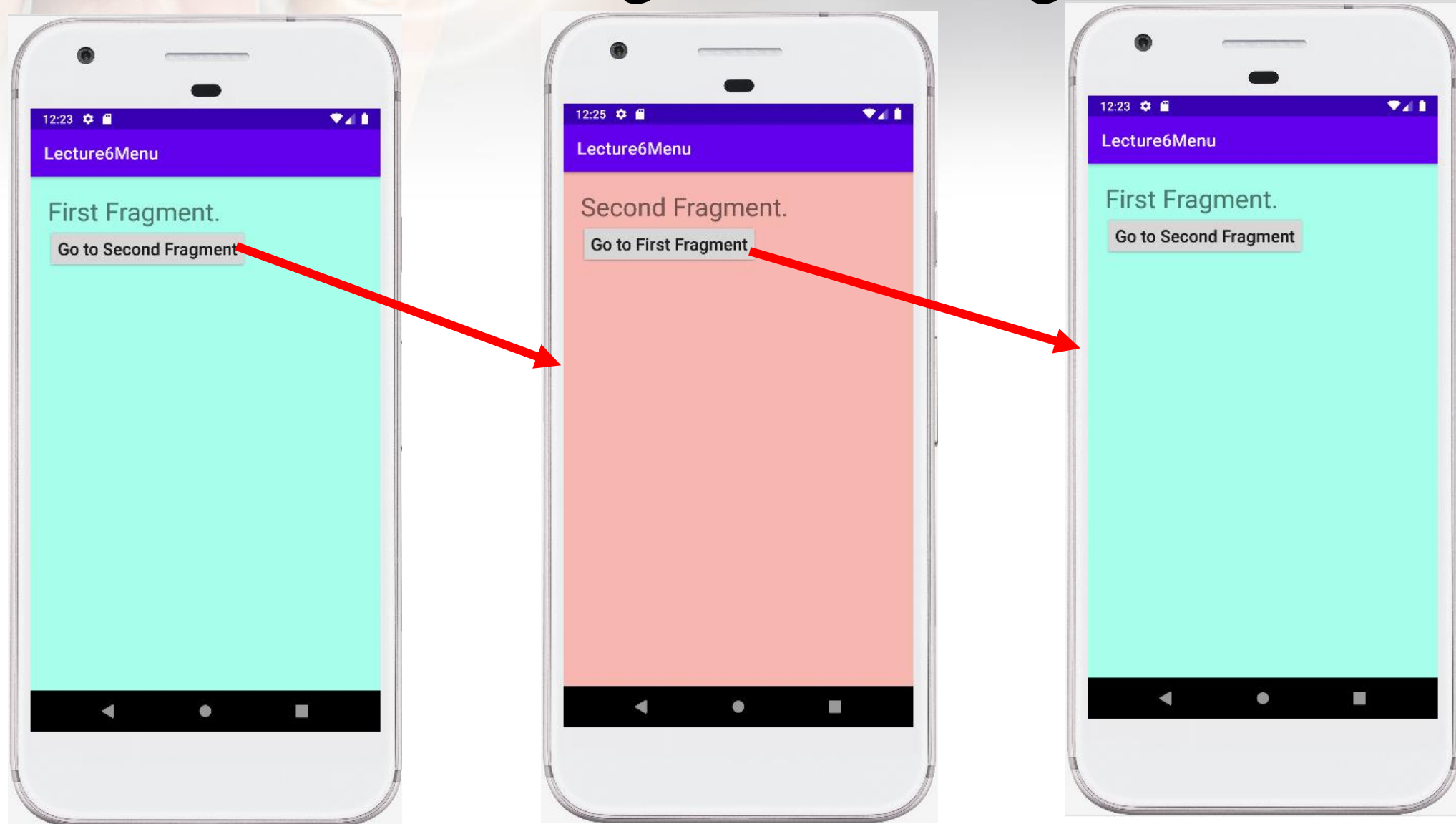
Call Another Fragment on Fragment

- activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        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"
        android:orientation="vertical"
        tools:context=".MainActivity">

  <FrameLayout
      android:id="@+id/frameLayout"
      android:layout_width="match_parent"
      android:layout_height="match_parent"/>

</LinearLayout>
```
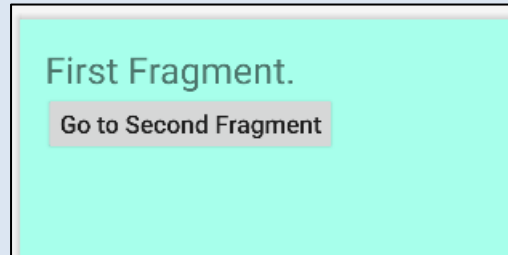
Call Another Fragment on Fragment

fragment_first.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:background="#A7FFEB"
    android:padding="20dp"
    tools:context=".FirstFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickFirstFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Go to Second Fragment" /> </LinearLayout>
```

fragment_second.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:background="#F8B6B1"
    android:padding="20dp"
    tools:context=".SecondFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Second Fragment."
        android:textSize="30sp"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClickSecondFrag"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:textSize="20sp"
        android:text="Go to First Fragment" /> </LinearLayout>
```
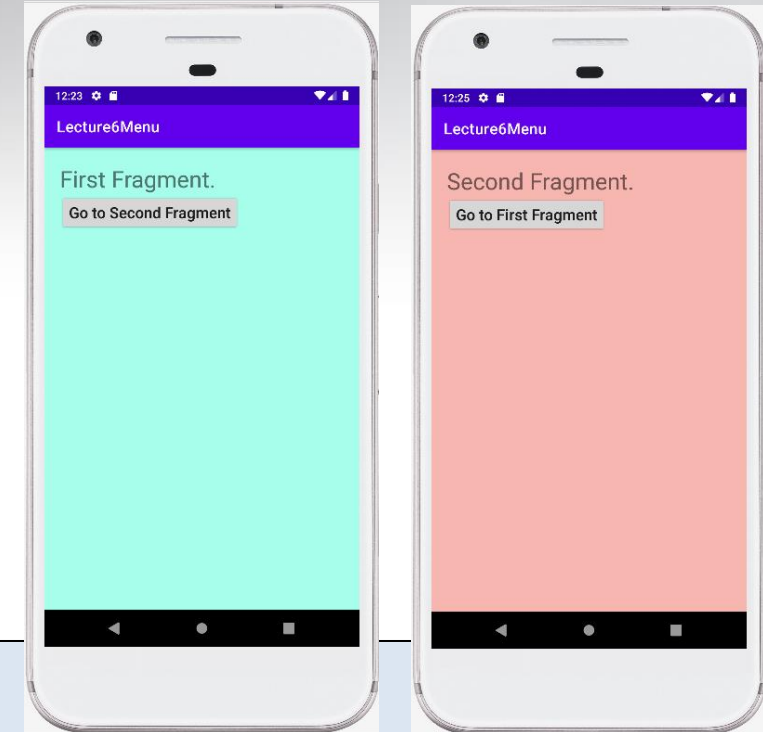
Call Another Fragment on Fragment

FirstFragment.kt

```kotlin
class FirstFragment : Fragment() {
    private lateinit var bindingFirst : FragmentFirstBinding
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        bindingFirst = FragmentFirstBinding.inflate(layoutInflater)
        bindingFirst.btnClickFirstFrag.setOnClickListener(){
            var fragment : Fragment? = null
            fragment = SecondFragment()
            replaceFragment(fragment)
        }
        return bindingFirst.root
    }
```

```kotlin
    fun replaceFragment(someFragment:Fragment){
        var binding: ActivityMainBinding
        binding = ActivityMainBinding.inflate(layoutInflater)
        val transaction = requireActivity().supportFragmentManager.beginTransaction()
        transaction.replace(binding.frameLayout.id, someFragment)
        transaction.addToBackStack(null)
        transaction.commit()
    }
}
```

54

Call Another Fragment on Fragment

SecondFragment.kt

```kotlin
class SecondFragment : Fragment() {
 private lateinit var bindingSecond : FragmentSecondBinding
   override fun onCreateView(
     inflater: LayoutInflater, container: ViewGroup?,
     savedInstanceState: Bundle?
   ): View? {
     // Inflate the layout for this fragment
     bindingSecond = FragmentSecondBinding.inflate(layoutInflater)
     bindingSecond.btnClickSecondFrag.setOnClickListener(){
        var fragment : Fragment? = null
        fragment = FirstFragment()
        replaceFragment(fragment)

     }
     return bindingSecond.root
 }
```

```kotlin
fun replaceFragment(someFragment:Fragment){
     var binding: ActivityMainBinding
     binding = ActivityMainBinding.inflate(layoutInflater)
     val transaction = requireActivity().supportFragmentManager.beginTransaction()
     transaction.replace(binding.frameLayout.id, someFragment)
     transaction.addToBackStack(null)
     transaction.commit()
   }
 }
```
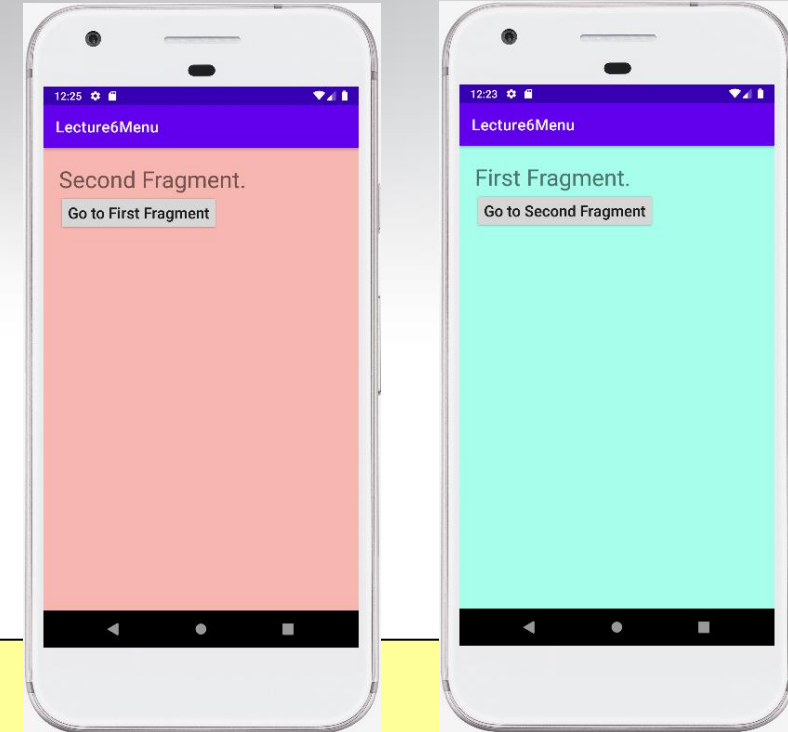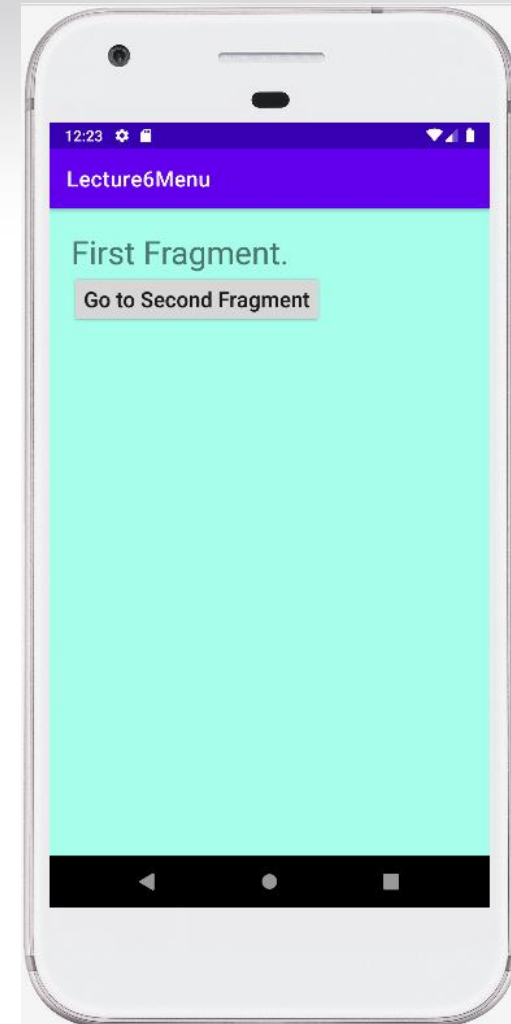
Call Another Fragment on Fragment

MainActivity.kt

```kotlin
class MainActivity : AppCompatActivity() {

    private lateinit var binding : ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)


        supportFragmentManager.beginTransaction().add(
            R.id.frameLayout,
            FirstFragment()
        ).commit()
    }
}
```


Lecture6Menu

First Fragment.
Go to Second Fragment

56

End of Chapter

References

- https://www.javatpoint.com/android-option-menu-example
- http://home.iitk.ac.in/~triveni/fragment.pptx
- https://marif.yolasite.com/resources/Android-Lecture6-Fragments.pptx
- http://www.cs.unibo.it/projects/android/2014/slides/16_fragments.ppt
- http://www.cse.bgu.ac.il/common/download.asp?FileName=Lecture%204.ppt&AppID=2&MainID=552&SecID=4667&MinID=3
- ftp://103.81.117.86/04%20IT%20Department/VBS/Android/android%20material/android%20material/UI.ppt
- https://abhiandroid.com/ui/fragment
- https://android--code.blogspot.com/2018/02/android-kotlin-menu-and.html