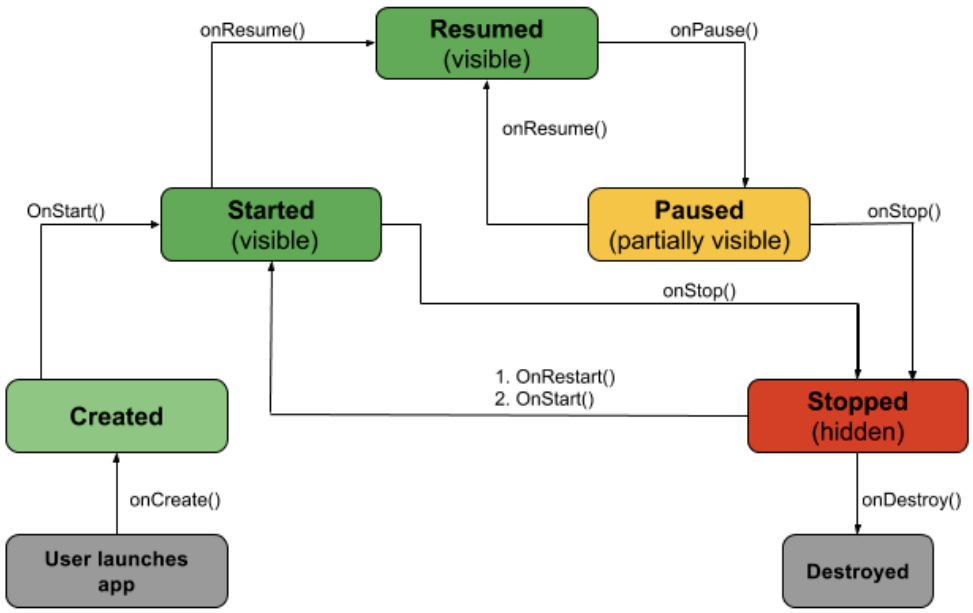
PRACTICAL 3

Programming Activities and fragments

Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments.

Activity Lifecycle:



- **onCreate():** Called by the OS when the activity is first created. This is where you initialize any UI elements or data objects. You also have the savedInstanceState of the activity that contains its previously saved state, and you can use it to recreate that state.\

```
fun onCreate(savedInstanceState: Bundle?) {
  super.onCreate(savedInstanceState)
  setContentView(R.layout.activity_task_description)
```

- **onStart():** Just before presenting the user with an activity, this method is called. It's always followed by onResume(). In here, you generally should start UI animations, audio based content or anything else that requires the activity's contents to be on screen.

-
- **onResume():** As an activity enters the foreground, this method is called. Here you have a good place to restart animations, update UI elements, restart camera previews, resume audio/video playback or initialize any components that you release during onPause().
- **onPause():** This method is called before sliding into the background. Here you should stop any visuals or audio associated with the activity such as UI animations, music playback or the camera. This method is followed by onResume() if the activity returns to the foreground or by onStop() if it becomes hidden.

- **onStop():** This method is called right after onPause(), when the activity is no longer visible to the user, and it's a good place to save data that you want to commit to the disk. It's followed by either onRestart(), if this activity is coming back to the foreground, or onDestroy() if it's being released from memory.

- **onRestart():** Called after stopping an activity, but just before starting it again. It's always followed by onStart().

- **onDestroy():** This is the final callback you'll receive from the OS before the activity is destroyed. You can trigger an activity's desctruction by calling finish(), or it can be triggered by the system when the system needs to recoup memory. If your activity includes any background threads or other long-running resources, destruction could lead to a memory leak if they're not released, so you need to remember to stop these processes here as well.

string.xml

```
<resources>
    <string name="app_name">State Change Activity</string>
    <string name="this_is_my_first_app">This is my first app!</string>
    <string name="click_second_activity">click second activity</string>
    <string name="click_third_activity">click third activity</string>
    <string name="this_is_another_activity">This is another Activity</string>
    <string name="this_is_first_activity">This is first Activity</string>
    <string name="mainactivity">MainActivity</string>
```

```
    <string name="open_second_activity">Open Second Activity</string>
    <string name="new_activity">New Activity</string>
    <string name="go_back_to_mainactivity">Go back to MainActivity</string>
</resources>
```

activity_main.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=".MainActivity">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="40dp"
    android:textColor="@android:color/holo_purple"
    android:text="@string/mainactivity"
    android:textSize="24sp"
    android:textStyle="bold" />

  <Button
    android:id="@+id/btnOpenAct2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:text="@string/open_second_activity" />
</RelativeLayout>
```

MainActivity.kt

```
package com.example.statechangeactivity

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    title = "KotlinApp"
    val button:Button = findViewById(R.id.btnOpenAct2)
    button.setOnClickListener {
```

```
        val intent = Intent(this@MainActivity, NewActivity::class.java)
        startActivity(intent)
    }
  }
}
```

activity_new.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=".NewActivity">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="40dp"
    android:text="@string/new_activity"
    android:textColor="@android:color/holo_green_dark"
    android:textSize="24sp"
    android:textStyle="bold" />

  <Button
    android:id="@+id/btnOpenMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:text="@string/go_back_to_mainactivity" />
</RelativeLayout>
```

NewActivity.kt

```
package com.example.statechangeactivity

import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class NewActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_new)
    title = "KotlinApp"; val button: Button = findViewById(R.id.btnOpenMain)
    button.setOnClickListener {
      val i = Intent(this@NewActivity, MainActivity::class.java)
      startActivity(i)
    }
```

```
    }
}
```

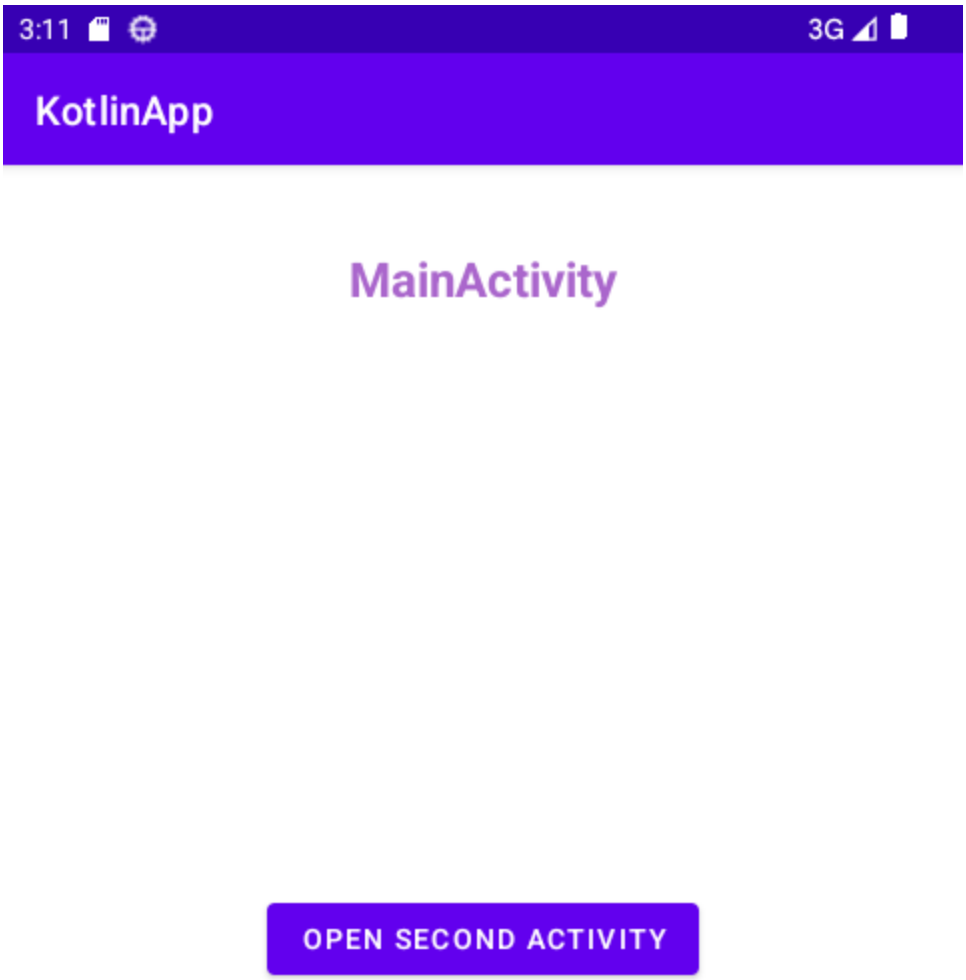
AndroidManifest.xml

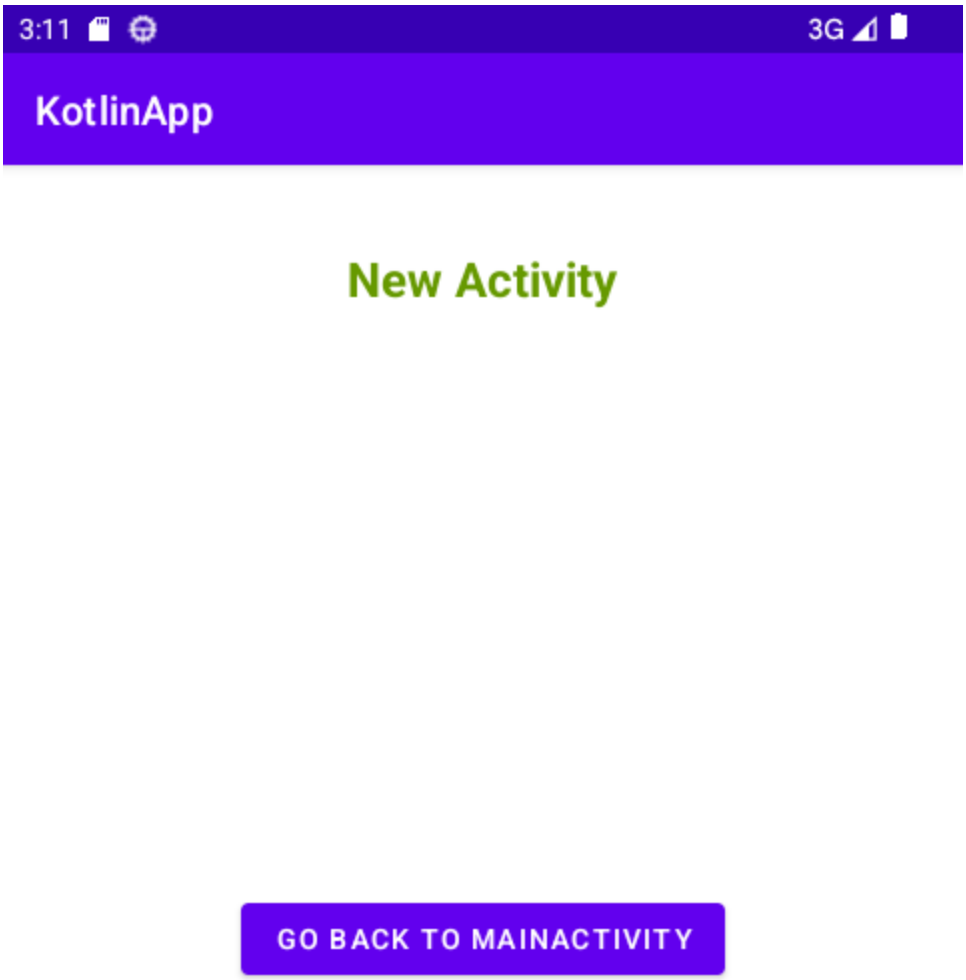
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.StateChangeActivity">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity android:name=".NewActivity"></activity>
    </application>
</manifest>
```

Output:


3:11
3G
KotlinApp
MainActivity
OPEN SECOND ACTIVITY


3:11
3G
KotlinApp
New Activity
GO BACK TO MAINACTIVITY