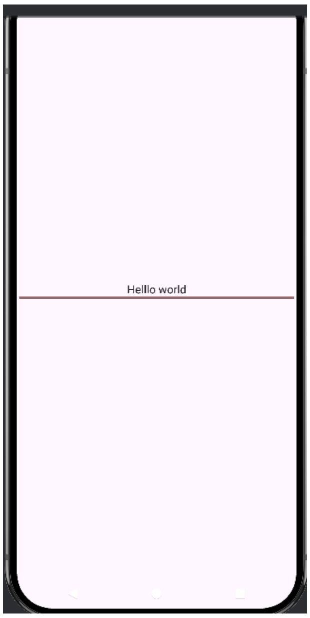
Android Programming

Practical 1:- 1. **Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals: Creating a Project, Android Components, Activities, Services, Content Providers, Broadcast Receivers, Interface overview, Creating Android Virtual device, USB debugging mode, Android Application Overview. Simple “Hello World” program.**

Activity_Main.Kt

```
 package com.aakash.hello
import android.support.v7.app.AppCompatActivity
 import android.os.Bundle
class MainActivity : AppCompatActivity() {
Override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main) } }
```



```
Helllo world
```

Activity_Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http помощью://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:layout_width="wrap_content"
android:text="Hello World!"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"/>
</android.support.constraint.ConstraintLayout>
```

BroadcastActivity:

Step 1: Create an Android App

Use Kotlin to build an Android app.

Step 2: Create a Broadcast Receiver

Extend `BroadcastReceiver` and override `onReceive()`.

```
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast

class MyReceiver : BroadcastReceiver() {
   override fun onReceive(context: Context, intent: Intent) {
      Toast.makeText(context, "Broadcast: Flight mode changed.", Toast.LENGTH_LONG).show()
   }
}
```

Step 3: Declare Broadcast Receiver in Manifest

Add `<receiver>` in `AndroidManifest.xml`.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="in.eyehunt.androidbroadcasts">

    <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/AppTheme">

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver android:name=".MyReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.AIRPLANE_MODE"/>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

Step 4: MainActivity Code (No Changes Needed)

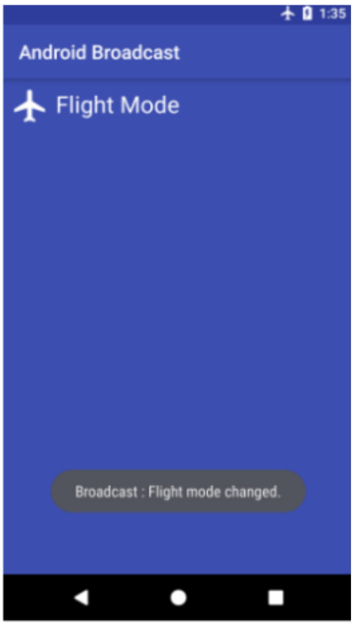
MainActivity.kt:

```
import android.support.v7.app.AppCompatActivity
import android.os.Bundle

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Step 5: Update `main_activity.xml`

Add `ImageView` & `TextView` to the layout.

```
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    tools:context="in.eyehunt.androidbroadcasts.MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="8dp"
        android:layout_marginTop="16dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/baseline_airplanemode_active_white_24" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="300dp"
        android:layout_height="36dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:gravity="center_vertical"
        android:text="Flight Mode"
        android:textColor="@color/colorWhite"
        android:textSize="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="@+id/imageView" />
</android.support.constraint.ConstraintLayout>
```



Broadcast : Flight mode changed.

Practical 2 :- Programming Resources Android Resources: (Color, Theme, String, Drawable, Dimension, Image),

Color.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#008577</color>
    <color name="colorPrimaryDark">#00574B</color>
    <color name="colorAccent">#D81B60</color>
</resources>
```

Theme:

Style.xml

Code

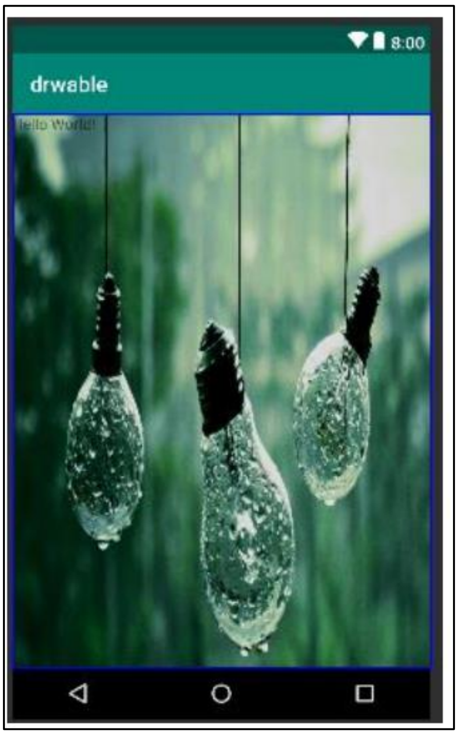
```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

String.xml

```
<resources>
    <string name="app_name">hello</string>
    <string name="numbers">
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </string>
</resources>
```

Drawable:

1. Right click on drawable folder
2. Copy the image if you want to create image drawable
3. Paste that image file inside the drawable folder



Practical 3:- Programming Activities and fragments
Activity Life Cycle, Activity methods, Multiple Activities, Life Cycle of fragments and multiple fragments.

Android Activity Lifecycle Callbacks

Android activities go through different lifecycle stages controlled by the OS. Below are the key lifecycle methods:

1□ onCreate()

📌 **Called when the activity is first created.**

✓□ Initialize UI components & data objects.

✓□ Restore the previous state using `savedInstanceState`.

2□ onStart()

📌 **Called before the activity becomes visible to the user.**

✓□ Start UI animations, audio, or other interactive elements.

✓□ Always followed by `onResume()`.

3□ onResume()

📌 **Called when the activity enters the foreground.**

✓□ Resume UI updates, animations, camera previews, and media playback.

✓□ Activity is now fully interactive.

4□ onPause()

📌 **Called when the activity is about to go into the background.**

✓□ Stop animations, audio, or camera previews.

✓□ Followed by `onResume()` if the activity returns or `onStop()` if it becomes invisible.

5□ onStop()

📌 **Called when the activity is no longer visible.**

✓□ Save data and release resources to improve performance.

✓□ Followed by either `onRestart()` (if returning) or `onDestroy()` (if ending).

6□ onRestart()

📌 **Called when restarting a stopped activity.**

✓□ Always followed by `onStart()`.

7□ onDestroy()

📌 **Called when the activity is being destroyed.**

✓□ Release background threads and resources to avoid memory leaks.

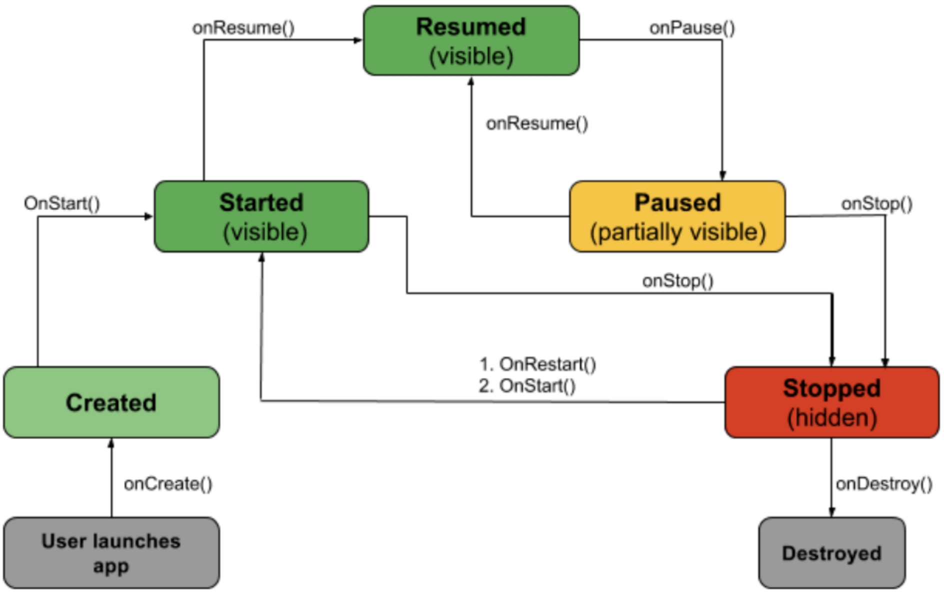
✓□ Triggered by calling `finish()` or when the system needs memory.

Lifecycle Flow:

`onCreate()` → `onStart()` → `onResume()` → (Activity running)
`onPause()` → `onStop()` → (`onRestart()` → `onStart()`) OR `onDestroy()`

This structure ensures proper resource management and a smooth user experience.

Activity Lifecycle:



Practical 4:- Programs related to different Layouts

Coordinate, Linear, Relative, Table, Absolute, Frame, List View, Grid View.

Android Layouts Overview

1□ **Linear Layout** (activity_main.xml)

Arranges elements **vertically** or **horizontally**.

Example: Three buttons to start, pause, and stop a service.

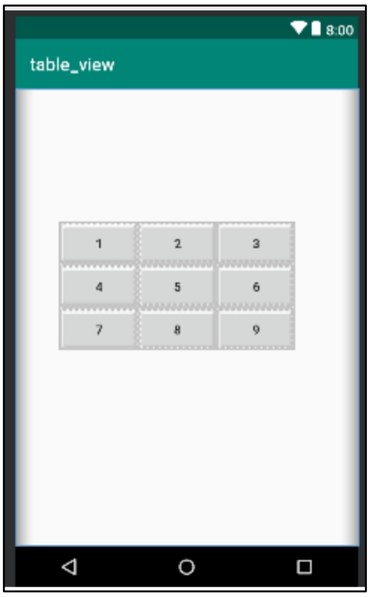
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical">

   <Button android:id="@+id/btnStartService"
      android:layout_width="270dp"
      android:layout_height="wrap_content"
      android:text="Start Service"/>

   <Button android:id="@+id/btnPauseService"
      android:layout_width="270dp"
      android:layout_height="wrap_content"
      android:text="Pause Service"/>

   <Button android:id="@+id/btnStopService"
      android:layout_width="270dp"
      android:layout_height="wrap_content"
      android:text="Stop Service"/>
</LinearLayout>
```

2□ Relative Layout (activity_main.xml)

Positions elements **relative to each other** or **the parent**.

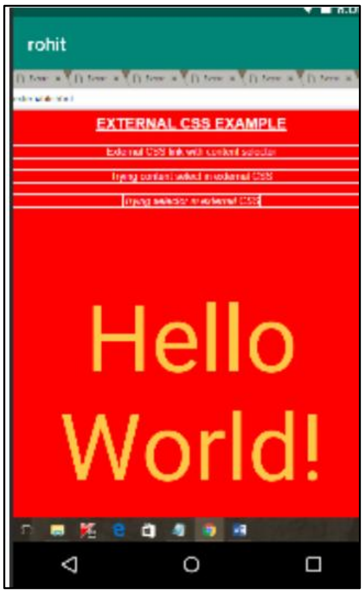
Example: EditText and two buttons inside a LinearLayout.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:paddingLeft="16dp"
   android:paddingRight="16dp">

   <EditText
      android:id="@+id/name"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:hint="@string/reminder" />

   <LinearLayout
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:layout_below="@+id/name">

      <Button android:id="@+id/button"
         android:layout_width="wrap_content"
```



8:00

table_view



rohit

EXTERNAL CSS EXAMPLE

External CSS link with content selector

Trying content select in external CSS

Trying selector in external CSS

Hello World!

```
            android:layout_height="wrap_content"
            android:text="Button 1"/>

        <Button android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 2"/>
    </LinearLayout>
</RelativeLayout>
```

3️⃣ Table Layout (activity_main.xml)

Arranges elements in a **grid-like table format**.

Example: Numeric keypad with buttons inside a `TableLayout`.

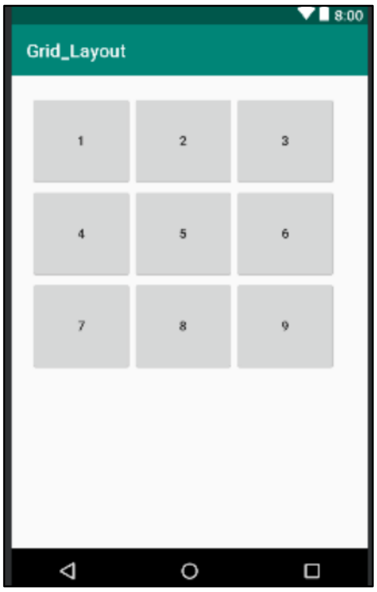
```
<TableLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="150dp">

    <TableRow>
        <Button android:id="@+id/btn1" android:text="1"/>
        <Button android:id="@+id/btn2" android:text="2"/>
        <Button android:id="@+id/btn3" android:text="3"/>
    </TableRow>

    <TableRow>
        <Button android:id="@+id/btn4" android:text="4"/>
        <Button android:id="@+id/btn5" android:text="5"/>
        <Button android:id="@+id/btn6" android:text="6"/>
    </TableRow>

    <TableRow>
        <Button android:id="@+id/btn7" android:text="7"/>
        <Button android:id="@+id/btn8" android:text="8"/>
        <Button android:id="@+id/btn9" android:text="9"/>
    </TableRow>
</TableLayout>
```

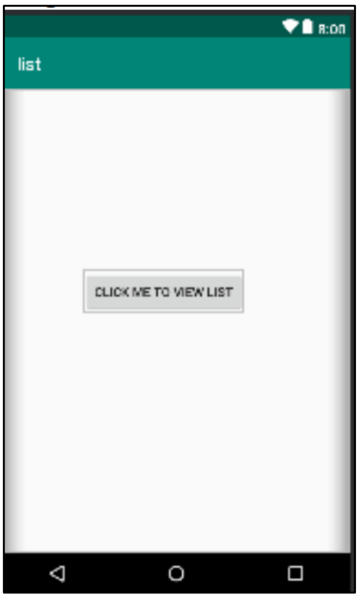
MainActivity.kt (Handles button clicks)

Displays a toast message when a button is clicked.

```
package com.r.table_view
import android.support.v7.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*
import org.jetbrains.anko.toast

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        btn1.setOnClickListener { toast("1") }
        btn2.setOnClickListener { toast("2") }
        btn3.setOnClickListener { toast("3") }
        btn4.setOnClickListener { toast("4") }
        btn5.setOnClickListener { toast("5") }
        btn6.setOnClickListener { toast("6") }
        btn7.setOnClickListener { toast("7") }
        btn8.setOnClickListener { toast("8") }
        btn9.setOnClickListener { toast("9") }
    }
}
```

Op:-





Practical 5 :- Programming UI elements

AppBar, Fragments, UI Components

MainActivity.kt
Handles **button clicks** to navigate to **LoginActivity** and **RegisterActivity**.

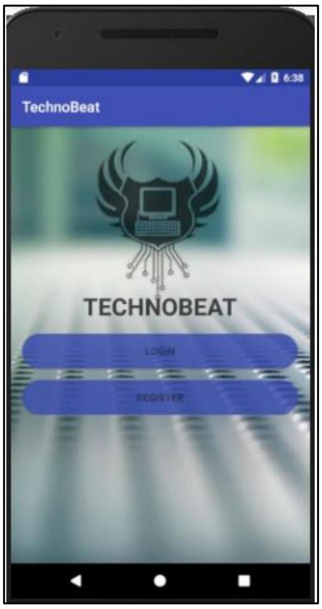
```
package rohit.technobeat

import android.content.Intent
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    login.setOnClickListener {
      startActivity(Intent(this, LoginActivity::class.java))
    }

    newaccount.setOnClickListener {
      startActivity(Intent(this, RegisterActivity::class.java))
    }
  }
}
```

activity_main.xml

ScrollView: Ensures **content is scrollable**.
LinearLayout: Arranges **title & buttons** vertically.
Buttons: Navigate to **Login** and **Register** screens.

```
<?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:gravity="center_horizontal"
  android:padding="@dimen/activity_vertical_margin"
  android:background="@drawable/home"
  tools:context=".MainActivity">

  <ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:gravity="center">

      <android.support.v7.widget.AppCompatTextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="210dp"
        android:alpha="0.7"
        android:text="TECHNOBEAT"
        android:textColor="#000000"
        android:textSize="33dp"
        android:textStyle="bold"/>

      <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="Login"
        android:background="@drawable/round_button"
        android:alpha="0.8"
        android:textStyle="bold" />

      <Button
        android:id="@+id/newaccount"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:text="REGISTER"
        android:background="@drawable/round_button"
        android:alpha="0.8"
        android:textStyle="bold" />
    </LinearLayout>
  </ScrollView>
</LinearLayout>
```

Op :-



Practical 6 :- Programming menus, dialog, dialog fragments

Kotlin AlertDialog (Precise & Clean)

```
val alertDialog = activity?.let {
   AlertDialog.Builder(it).apply {
      setPositiveButton(R.string.ok) { _, _ -> /* OK button clicked */ }
      setNegativeButton(R.string.cancel) { _, _ -> /* Cancel button clicked */ }
      // Set other dialog properties if needed
   }.create()
}
```

menu.xml *(Defines the menu items)*

Menu items with **icons** and **titles**.

`showAsAction="always"` ensures `menu_1` is **always visible** in the toolbar.

```
<?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/menu_1"
      android:icon="@drawable/ic_menu_1"
      android:title="Menu 1"
      app:showAsAction="always" />

   <item
      android:id="@+id/menu_2"
      android:icon="@drawable/ic_menu_2"
      android:title="Menu 2" />

   <item
      android:id="@+id/menu_3"
      android:icon="@drawable/ic_menu_3"
      android:title="Menu 3" />

   <item
      android:id="@+id/menu_4"
      android:icon="@drawable/ic_menu_4"
      android:title="Menu 4" />
</menu>
```

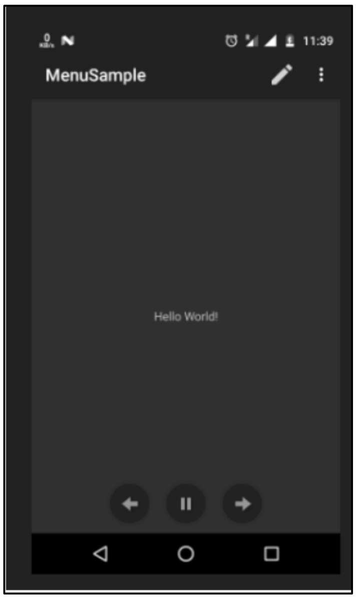
MainActivity.kt (Handles menu selection)

Inflates menu in `onCreateOptionsMenu()`.

Handles item clicks in `onOptionsItemSelected()` by showing a **Toast message**.

```
package rohit.com

import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast

class MainActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)
   }

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

   override fun onOptionsItemSelected(item: MenuItem): Boolean {
      val message = when (item.itemId) {
         R.id.menu_1 -> "Menu 1 is selected"
         R.id.menu_2 -> "Menu 2 is selected"
         R.id.menu_3 -> "Menu 3 is selected"
         R.id.menu_4 -> "Menu 4 is selected"
         else -> return super.onOptionsItemSelected(item)
      }
      Toast.makeText(this, message, Toast.LENGTH_SHORT).show()
      return true
   }
}
```

Op :-



Practical 7 :- Programs on Intents, Events, Listeners and Adapters
The Android Intent Class, Using Events and Event Listeners

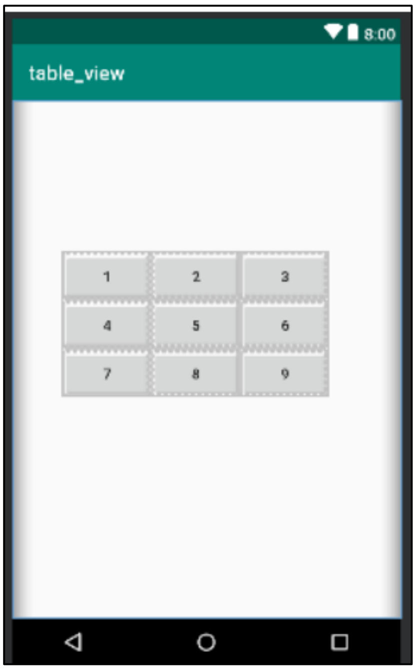
activity_main.xml

```
<?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"
  tools:context=".MainActivity">

  <TableLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_marginTop="150dp">

    <TableRow>
      <Button android:id="@+id/btn1" android:text="1"/>
      <Button android:id="@+id/btn2" android:text="2"/>
      <Button android:id="@+id/btn3" android:text="3"/>
    </TableRow>
    <TableRow>
      <Button android:id="@+id/btn4" android:text="4"/>
      <Button android:id="@+id/btn5" android:text="5"/>
      <Button android:id="@+id/btn6" android:text="6"/>
    </TableRow>
    <TableRow>
      <Button android:id="@+id/btn7" android:text="7"/>
      <Button android:id="@+id/btn8" android:text="8"/>
      <Button android:id="@+id/btn9" android:text="9"/>
    </TableRow>

  </TableLayout>
</LinearLayout>
```

MainActivity.kt

```
package com.r.table_view
import android.os.Bundle
import android.support.v7.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*
import org.jetbrains.anko.toast

class MainActivity : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    listOf(btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9).forEachIndexed { index, button ->
button.setOnClickListener { toast("${index + 1}") }}}}
```

Op:-



Prcatical 8 :- Programs on Services, notification and broadcast receivers

1. Services in Kotlin

Services are background tasks in Android. Common service-related components include:
IntentService – Handles asynchronous tasks.
onStartCommand() – Called when a service starts.
onHandleIntent() – Processes intents sequentially.

2. Notifications & Broadcast Receivers

Step 1: Create an Android App
Follow a Kotlin Android tutorial to set up a basic project.
Step 2: Create a Broadcast Receiver
Create `MyReceiver.kt` to listen for broadcasted intents.

MyReceiver.kt

```
package `in`.eyehunt.androidbroadcasts

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast

class MyReceiver : BroadcastReceiver() {
   override fun onReceive(context: Context, intent: Intent) {
      Toast.makeText(context, "Broadcast: Flight mode changed.",
Toast.LENGTH_LONG).show()
   }
}
```

Step 3: Declare Receiver in Manifest

Modify `AndroidManifest.xml` to register the receiver.

AndroidManifest.xml

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

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:theme="@style/AppTheme">

      <activity android:name=".MainActivity">
         <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
         </intent-filter>
      </activity>

      <receiver
```

```
            android:name=".MyReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.AIRPLANE_MODE"/>
            </intent-filter>
        </receiver>

    </application>
</manifest>
```

Step 4: MainActivity Code

No additional logic needed; just set the content view.

MainActivity.kt

```
package `in`.eyehunt.androidbroadcasts

import android.os.Bundle
import android.support.v7.app.AppCompatActivity

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

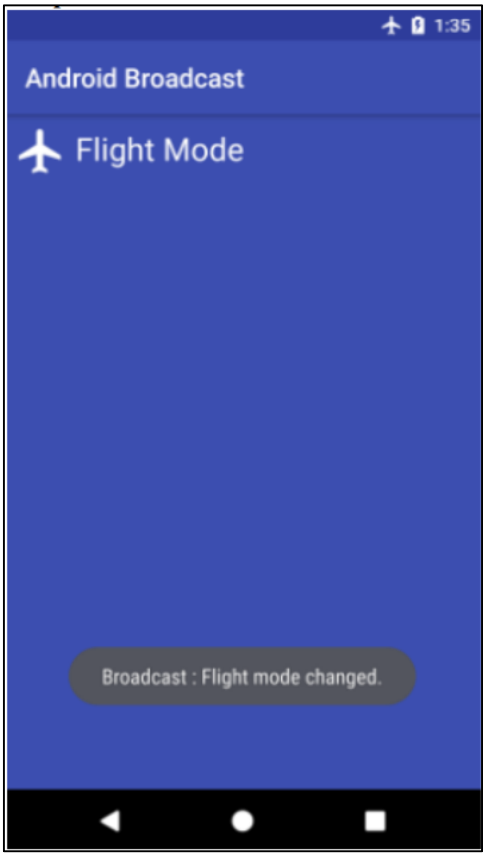
Step 5: Update Layout (main_activity.xml)

Add an **ImageView** and **TextView** to display flight mode status.

main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorPrimary"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="8dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@mipmap/baseline_airplanemode_active_white_24" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="300dp"
        android:layout_height="36dp"
        android:gravity="center_vertical"
        android:text="Flight Mode"
        android:textColor="@color/colorWhite"
        android:textSize="24dp"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="@+id/imageView" />
</android.support.constraint.ConstraintLayout>
```

Op:-



Broadcast : Flight mode changed.