# INDEX

| Sr. No | Practical | Title | Page No | Date | Signature |
|---|---|---|---|---|---|
| 1 | **Practical 1** | **Introduction to Android, Introduction to Android Studio IDE, Application Fundamentals:**<br>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. | | | |
| 2 | **Practical 2** | **Programming Resources**<br>Android Resources: (Color, Theme, String, Drawable, Dimension, Image). | | | |
| 3 | **Practical 3** | **Programming Activities and fragments**<br>Activity Life Cycle | | | |
| 4 | **Practical 4** | **Programs related to different Layouts**<br>Coordinate, Linear, Relative, Table, Absolute, Frame, List View, Grid View | | | |
| 5 | **Practical 5** | **Programming UI elements**<br>App Bar, Fragments, UI Components | | | |

# INDEX

| Sr. No | Practical | Title | Page No | Date | Signature |
|---|---|---|---|---|---|
| 6 | **Practical 6** | **Design an application representing a simple calculator.** | | | |
| 7 | **Practical 7** | **Programming menus, dialog, dialog fragments**<br>Develop an application for working with Menus and Screen Navigation. | | | |
| 8 | **Practical 8** | **Programs on services, notification and broadcast services**<br>Develop an application for working with notifications | | | |
| 9 | **Practical 9** | **Database Programming with SQLite**<br>Design a simple to-do list application using SQLite | | | |
| 10 | **Practical 10** | **Programming on Intents , Events , Listeners and Adapters**<br>The Android Intent Class, Using Events and Events Listeners | | | |

# Practical 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.

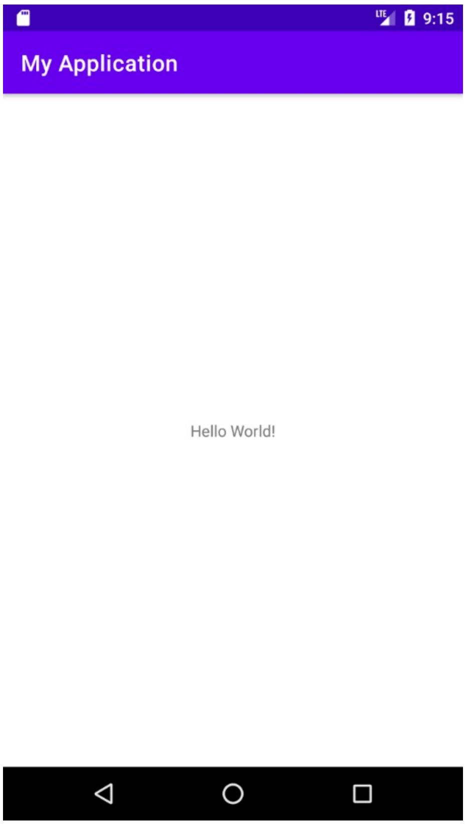
### main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".MainActivity"
    android:background="@drawable/images">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World This is an advanced Mobile Programming!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Practical1.java

```
packagecom.example.practical1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity

 {

    @Override

    protected void onCreate(Bundle savedInstanceState)

{

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

}
```

**OUTPUT**



My Application

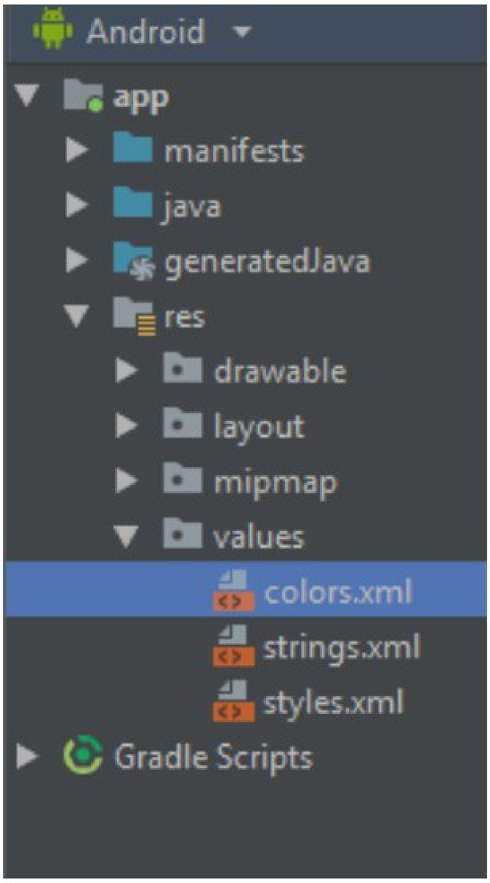
Hello World!

# Practical No.2

## Programming Resources

## Android Resources: (Color, Theme, String, Drawable, Dimension, Image).

## Color:



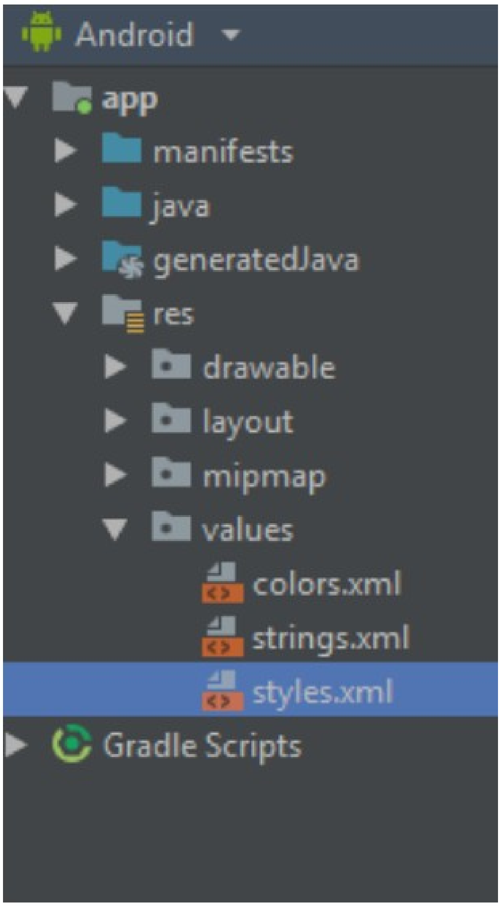
**Color.xml**

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

<resources>

<color name="colorPrimary">#008577</color>

<color name="colorPrimaryDark">#00574B</color>

<color name="colorAccent">#D81B60</color>

</resources>
```

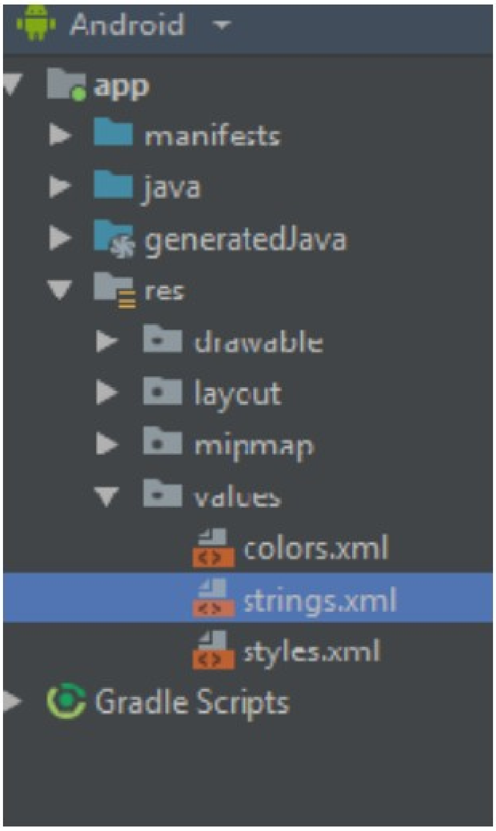
**Theme**:

**Style.xml**

```
<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:

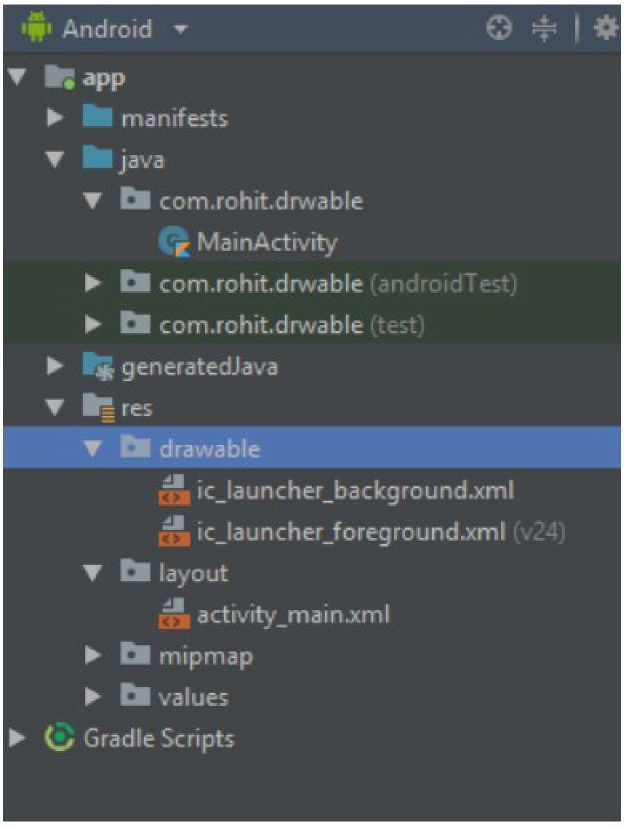


**String.xml**

```
<resources>

<string name="app_name">hello</string>

<string name="numbers">

<item>1</item>

<item>2</item>

<item>3</item>

</item>

</string>

</resources>
```

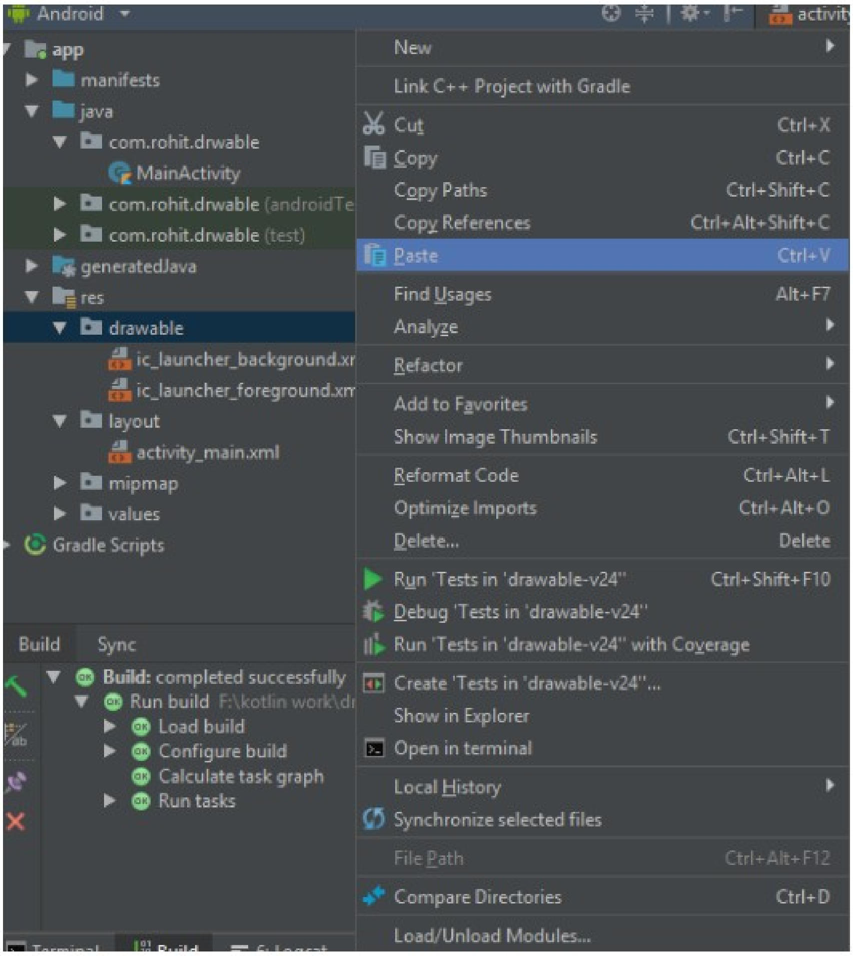
Drawable:

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


Android
activity
app
manifests
java
com.rohit.drwable
MainActivity
com.rohit.drwable (androidTe
com.rohit.drwable (test)
generatedJava
res
drawable
ic_launcher_background.xr
ic_launcher_foreground.xm
layout
activity_main.xml
mipmap
values
Gradle Scripts
Build
Sync
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy Ctrl+C
Copy Paths Ctrl+Shift+C
Copy References Ctrl+Alt+Shift+C
Paste Ctrl+V
Find Usages Alt+F7
Analyze
Refactor
Add to Favorites
Show Image Thumbnails Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Run 'Tests in 'drawable-v24'' Ctrl+Shift+F10
Debug 'Tests in 'drawable-v24''
Run 'Tests in 'drawable-v24'' with Coverage
Create 'Tests in 'drawable-v24''...
Show in Explorer
Open in terminal
Local History
Synchronize selected files
File Path Ctrl+Alt+F12
Compare Directories Ctrl+D
Load/Unload Modules...

```
Build: completed successfully
  Run build F:\kotlin work\dr
    Load build
    Configure build
    Calculate task graph
    Run tasks
```

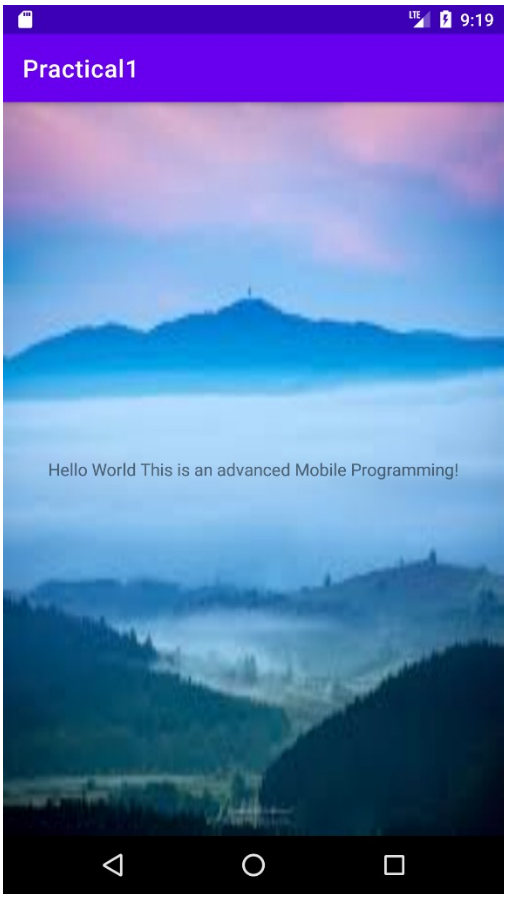
**main_activity.xml**

```
<?xml   version="1.0"encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content
        android:layout_height="wrap_content"
        android:text="Hello World This is an advanced Mobile Programming!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

**Practical2.java**

```
package com.example.practical1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity
{

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

## OUTPUT



Practical1

Hello World This is an advanced Mobile Programming!

# 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.

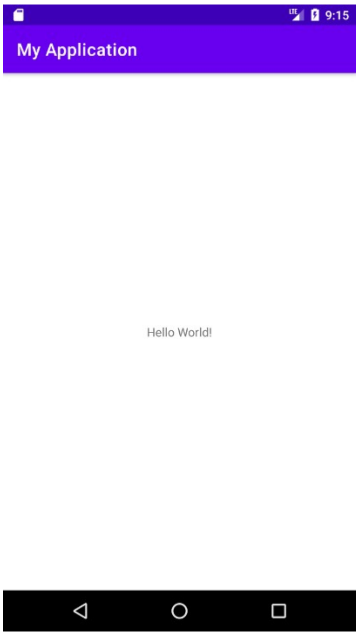
**activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

**MainActivity.java**

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity
{

    String tag = "Activity Life Cycle";
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(tag, "onCreate() Method");
    }

    @Override
    protected void onStart() {
        super.onStart();
        Log.d(tag ,"onStart() Method");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d(tag,"onRestart() Method");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.d(tag,"onPause() Method");
    }

    @Override
    protected void onStop() {
        super.onStop();
        Log.d(tag,"onStop() Method");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(tag,"onDestroy() Method");
    }
}
```

**OUTPUT**



```
02/13 08:45:24: Launching 'app' on Pixel 2 API 24.

App restart successful without requiring a re-install.

$ adb shell am start -n "com.example.myapplication/com.example.myapplication.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER

Connected to process 4592 on device 'Pixel_2_API_24 [emulator-5554]'.
```

```
Capturing and displaying logcat messages from application. This behavior can be disabled in the "Logcat output" section of the "Debugger" settings page.

I/art: Late-enabling -Xcheck:jni

W/art: Unexpected CPU variant for X86 using defaults: x86

W/System: ClassLoader referenced unknown path: /data/app/com.example.myapplication-1/lib/x86

W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter androidx.vectordrawable.graphics.drawable.VectorDrawableCompat.updateTintFilter(android.graphics.PorterDuffColorFilter, android.content.res.ColorStateList, android.graphics.PorterDuff$Mode) would have incorrectly overridden the package-private method in android.graphics.drawable.Drawable

D/Activity Life Cycle: onCreate() Method

D/Activity Life Cycle: onStart() Method

D/: HostConnection::get() New Host Connection established 0xa6c32d40, tid 4592

D/: HostConnection::get() New Host Connection established 0xa6c4e700, tid 4607

I/OpenGLRenderer: Initialized EGL, version 1.4

D/OpenGLRenderer: Swap behavior 1

W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED, retrying without...

D/OpenGLRenderer: Swap behavior 0

D/EGL_emulation: eglCreateContext: 0xa8205060: maj 2 min 0 rcv 2

D/EGL_emulation: eglMakeCurrent: 0xa8205060: ver 2 0 (tinfo 0xa8203510)

D/EGL_emulation: eglMakeCurrent: 0xa8205060: ver 2 0 (tinfo 0xa8203510)

D/Activity Life Cycle: onPause() Method

D/EGL_emulation: eglMakeCurrent: 0xa8205060: ver 2 0 (tinfo 0xa8203510)

D/Activity Life Cycle: onStop() Method

D/Activity Life Cycle: onRestart() Method

D/Activity Life Cycle: onStart() Method
```

# Practical 4

## Programs related to different Layouts

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

1. Linear Layout

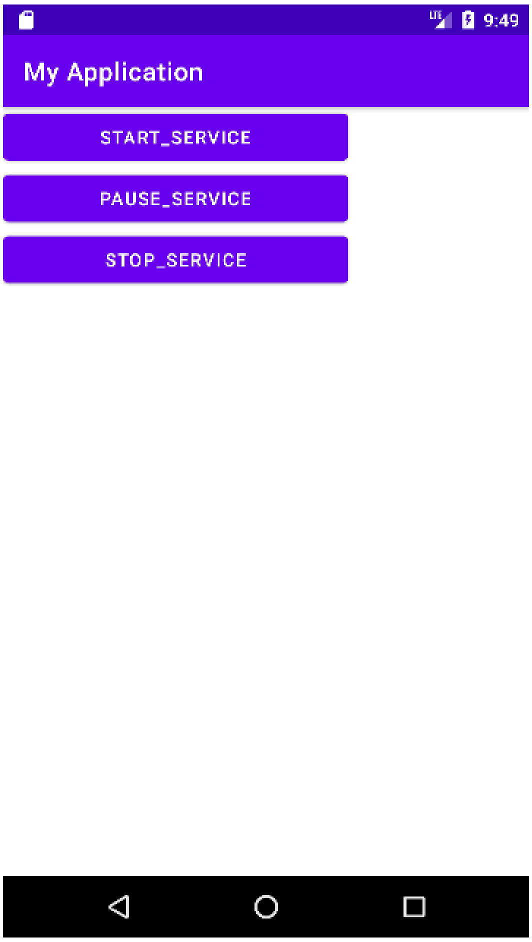
**LinearLayout.xml**

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

# MainActivity.java

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
   {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }

}
```

## OUTPUT

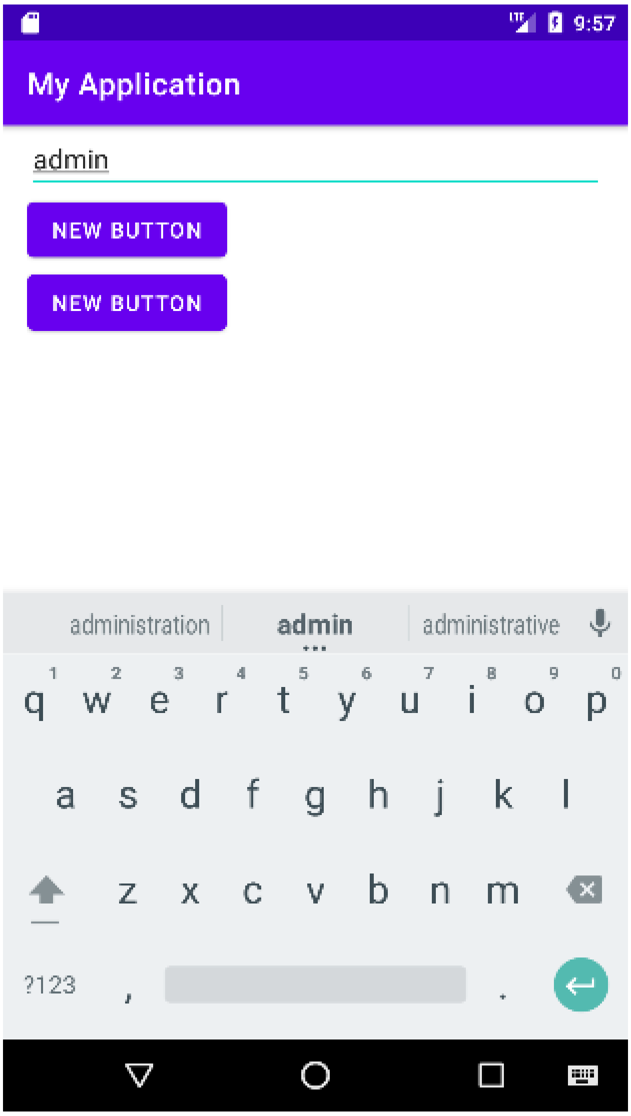


## 2. Relative Layout

## Relative Layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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_alignParentStart="true"
        android:layout_below="@+id/name">
        <Button
            android:layout_width="wrap_content
            android:layout_height="wrap_content"
            android:text="New Button
            android:id="@+id/button" />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="New Button"
            android:id="@+id/button2" />
    </LinearLayout>
</RelativeLayout>
```

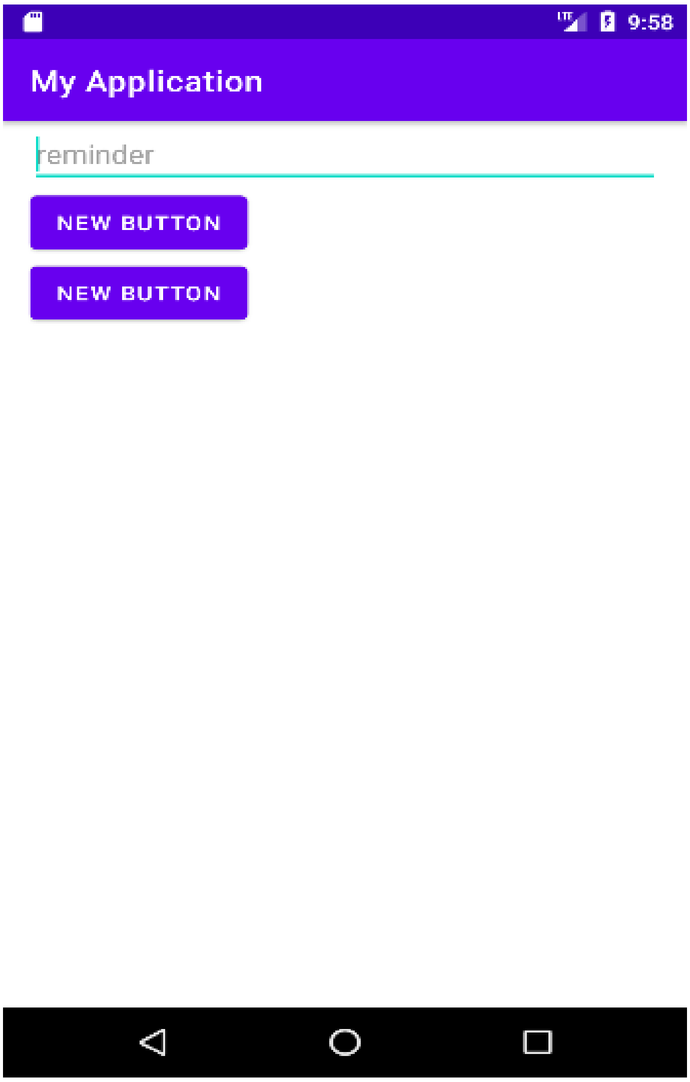
**OUTPUT**



9:57

My Application

admin


9:58
My Application
reminder
NEW BUTTON
NEW BUTTON

3.Frame Layout

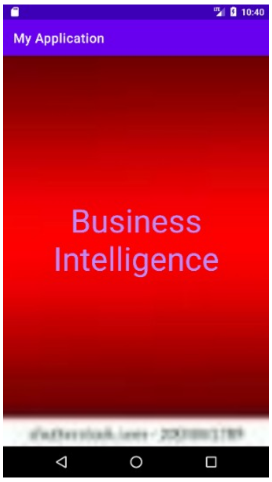
**Framelayout.xml**

```
<?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=".MainActivity">
    <ImageView android:layout_width="match_parent
        android:layout_height="match_parent"
        android:src="@drawable/red"
        android:scaleType="centerCrop"/>
    <TextView
        android:textSize="50dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Business Intelligence"
        android:gravity="center"
        android:textColor="#FFBB86FC"
        android:layout_marginTop="220dp"
        />
</FrameLayout>
```

**MainAciivity.java**

```
package com.example.myapplication
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }

}
```

**OUTPUT**



4 Grid Layout

***GridLayout.xml***

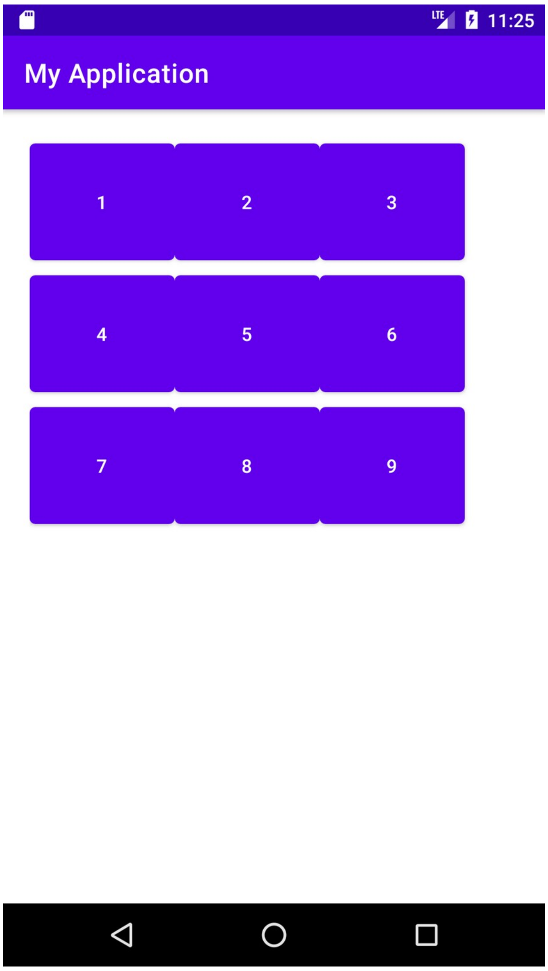
```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:rowCount="3"
    android:columnCount="3"
    android:padding="20dp">
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="1"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="2"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="3"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="4"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="5"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="6"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="7"/>
    <Button
```

```
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="8"/>
    <Button
        android:layout_width="110dp"
        android:layout_height="100dp"
        android:text="9"/>
</GridLayout>
```

**MainActivity.java**

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

## OUTPUT



5 List View

## Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
    <Button
        android:id="@+id/btn"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginStart="90dp"
         android:layout_marginTop="200dp"
         android:text="@string/click_me_to_view_list" />
</LinearLayout>
```

## Strings.xml

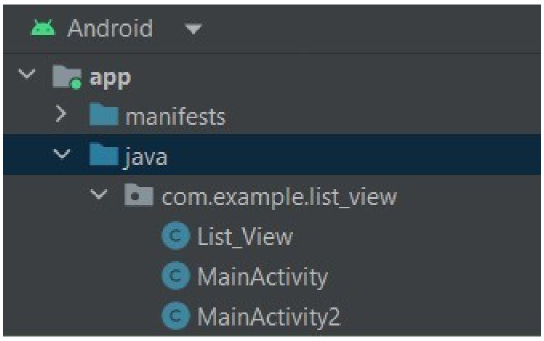
```
<resources>
    <string name="app_name">ListView</string>
    <string name="click_me_to_view_list">Click me to view list</string>
    <string-array name="array_technology">
        <item>Android</item>
        <item>Java</item>
        <item>Php</item>
        <item>Hadoop</item>
        <item>Sap</item>
        <item>Python</item>
        <item>Ajax</item>
        <item>C++</item>
        <item>Ruby</item>
        <item>Rails</item>
        <item>.Net</item>
        <item>Perl</item>
    </string-array>
</resources>
```

## MainActivity.java

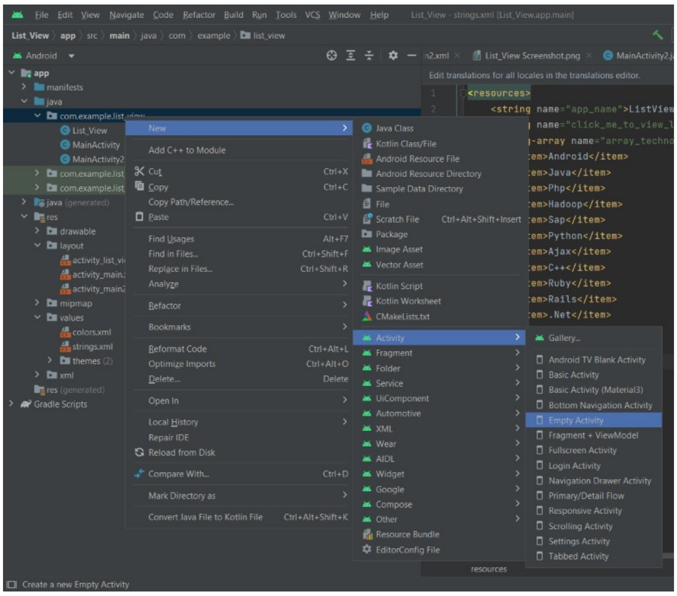
```
package com.example.list_view;
import android.content.Intent;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity
{
   Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button =(Button) findViewById(R.id.btn);
        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
               startActivity(new Intent(MainActivity.this,MainActivity2.class));
            }
        });
    }
}
```

## Steps to create activity(java) file:

## 1]Right click on com.example.package_name  folder



## 2]New>Activity>Empty Activity



```
<resources>
    <string name="app_name">ListView
    name="click_me_to_view_l
-array name="array_techno
em>Android</item>
em>Java</item>
em>Php</item>
em>Hadoop</item>
em>Sap</item>
em>Python</item>
em>Ajax</item>
em>C++</item>
em>Ruby</item>
em>Rails</item>
em>.Net</item>
```

## 3]Give the name to the activity  MainActivity2.java (you can change it if u want to )

```
package com.example.list_view;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
public class MainActivity2 extends AppCompatActivity {
    ListView listView;
    String [] listItem;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        listView = (ListView) findViewById(R.id.simpleListView);
        listItem = getResources().getStringArray(R.array.array_technology);
        final ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, android.R.id.text1, listItem);
        listView.setAdapter(arrayAdapter);
        listView.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long
l) {
                // TODO Auto-generated method stub
                String value=arrayAdapter.getItem(position);
                Toast.makeText(getApplicationContext(),value,Toast.LENGTH_SHORT).show();

            }
        });

    }
}
```

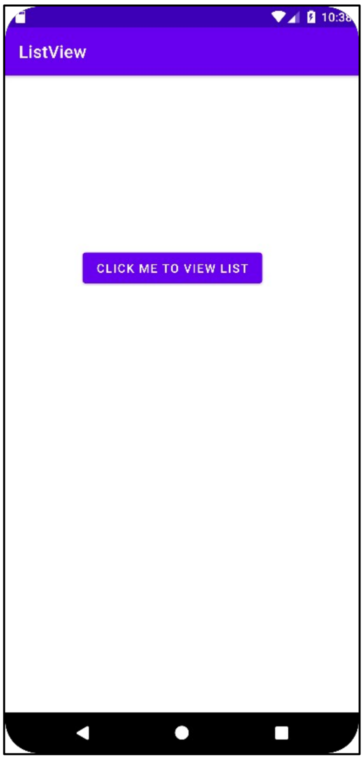
## With activity a xml file would be created as well.

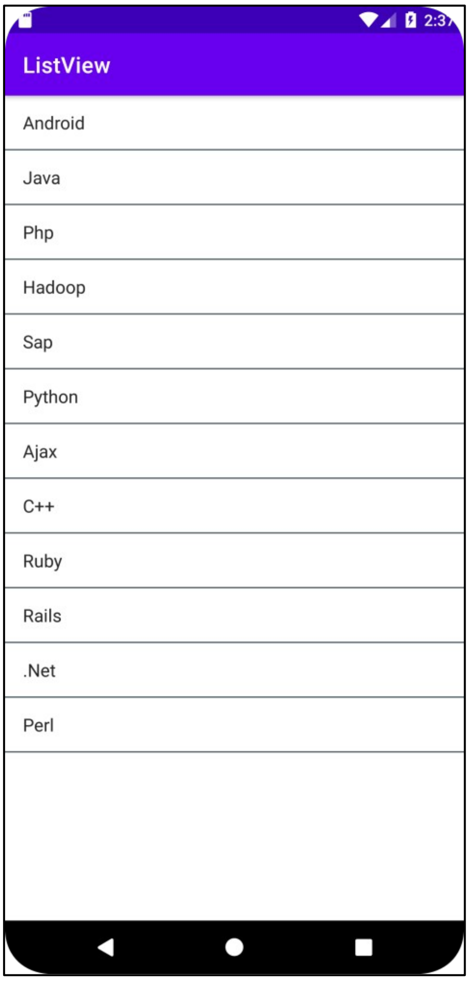
## Remove the extra part, copy paste given code below.

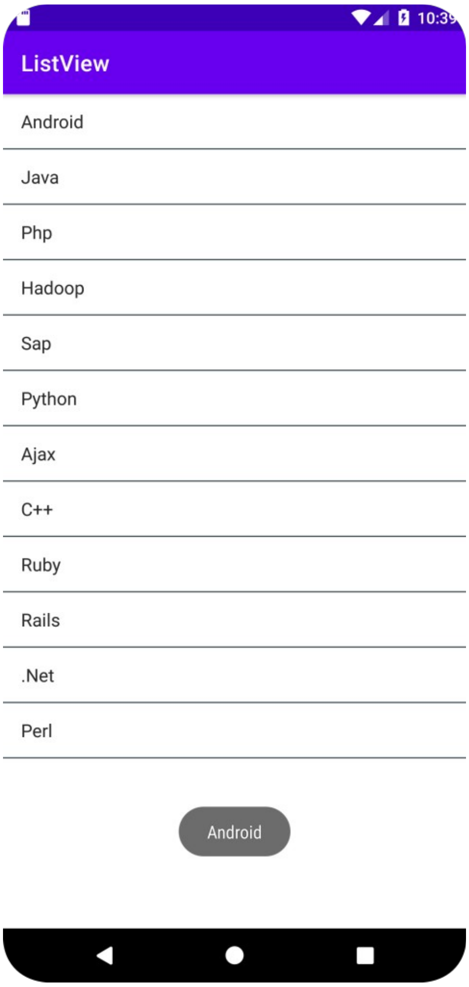
## Activity_main2.xml

```
<?xml version="1.0" encoding="utf-8"?>
    <ListView xmlns:android="http://schemas.android.com/apk/res/android"
      android:id="@+id/simpleListView"
      android:layout_width="match_parent"
      android:layout_height="731dp"
        android:divider="@color/material_blue_grey_800"
        android:dividerHeight="1dp" >
 </ListView>
```

## Output:




2:37
ListView
Android
Java
Php
Hadoop
Sap
Python
Ajax
C++
Ruby
Rails
.Net
Perl


10:39
ListView
Android
Java
Php
Hadoop
Sap
Python
Ajax
C++
Ruby
Rails
.Net
Perl
Android

# Practical 5

# Programming UI elements

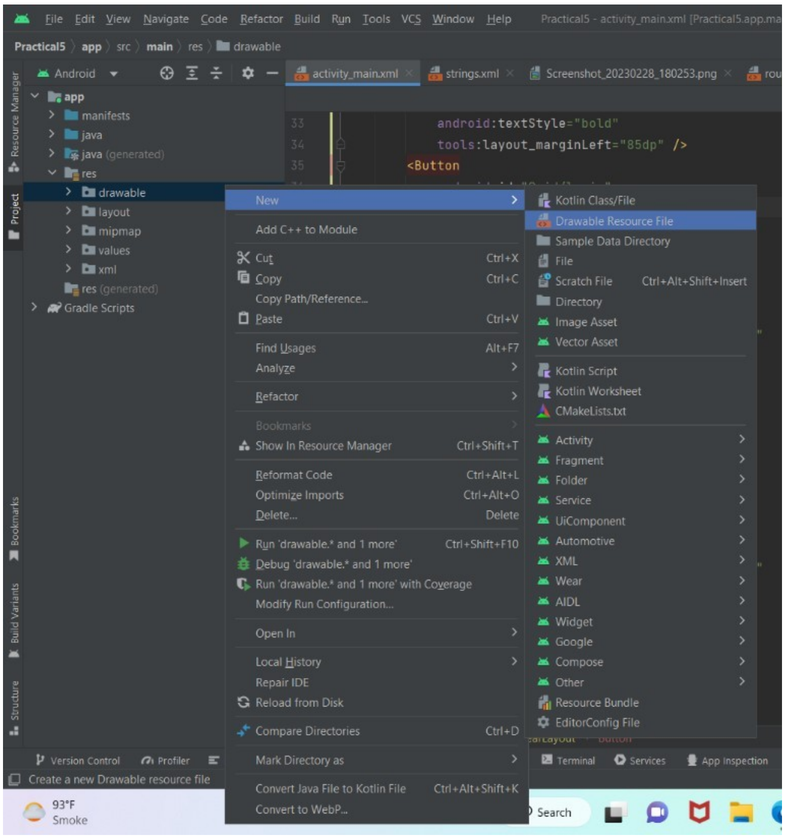
Design App With UI:

**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"
android:gravity="center_horizontal"
android:orientation="vertical"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:background="@drawable/scenery"
tools:context=".MainActivity">

    <ScrollView
        android:id="@+id/login_form"
        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">
        <androidx.appcompat.widget.AppCompatTextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginTop="210dp"
         android:alpha="0.7"
         android:text="@string/Nature"
         android:textColor="#000000"
         android:textSize="33dp"
         android:textStyle="bold"
            tools:layout_marginLeft="85dp" />
        <Button
            android:id="@+id/login"
            style="?android:textAppearanceSmall"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="@string/login"
            android:textColor="#FFFF"
            android:background="@drawable/round_button"
            android:alpha="0.8"
            android:textStyle="bold" />
        <Button
            android:id="@+id/new_account"
            style="?android:textAppearanceSmall"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:text="@string/register"
            android:textColor="#FFFF"
            android:background="@drawable/round_button"
            android:alpha="0.8"
            android:textStyle="bold" />
    </LinearLayout>
</ScrollView>
</LinearLayout>
```

# Steps to customize the button:

## 1]Go to res > drawable>new drawable resource 2]Give name to the file.



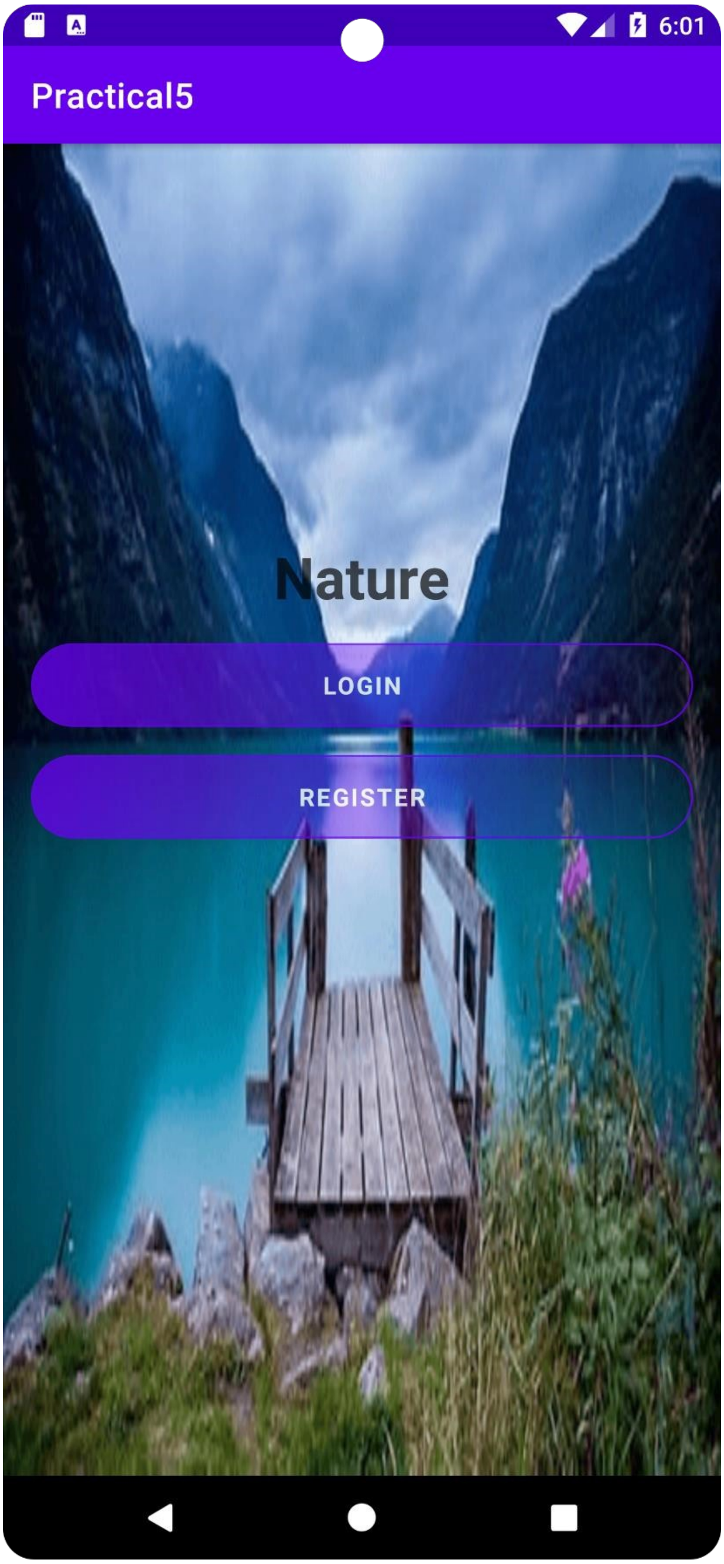
```
android:textStyle="bold"
tools:layout_marginLeft="85dp" />
<Button
```

Create a new Drawable resource file

## round_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:color="#388e3c"
        android:width="1dp"/>
    <corners
        android:radius="190dp"/>
    <solid
        android:color="#2e7d32"/>
    <gradient
        android:startColor="#e0f7fa"/>
</shape>
```

# Output



# Practical 6

# Design an application representing a simple calculator.

**XML Code:**

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="wrap_content"
tools:context="ty.practical1.MainActivity">
   <TextView
       android:id="@+id/txtDisplay"
android:layout_width="match_parent"
android:layout_height="90dp"
android:maxLength="15"
android:paddingLeft="10sp"
android:textAppearance="@android:style/TextAppearance.
DeviceDefault.Large”android:textSize="40sp" />

 <LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:layout_below="@+id/txtDisplay"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true">
  <LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">
<Button
android:id="@+id/btnSeven"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="7"
android:textSize="30dp"/>

 <Button
android:id="@+id/btnEight"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="8"
 android:textSize="30dp"/>
```

```
<Button
android:id="@+id/btnNine"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="9"
android:textSize="30dp"/>

    <Button
    android:id="@+id/btnDivide"
     android:layout_width="90dp"

android:layout_height="90dp"

android:text="/"
android:textSize="30dp"/>

      </LinearLayout>
      <LinearLayout
android:layout_width="match_parent"
 android:layout_height="wrap_content"
android:orientation="horizontal">
         <Button
android:id="@+id/btnFour"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="4"
android:textSize="30dp"/>

         <Button
android:id="@+id/btnFive"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="5"
android:textSize="30dp"/>

         <Button
android:id="@+id/btnSix"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="6"
android:textSize="30dp"/>

         <Button
android:id="@+id/btnMultiply"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="*"
android:textSize="30dp"/>

      </LinearLayout>
      <LinearLayout
 android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

            <Button
android:id="@+id/btnOne"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="1"
android:textSize="30dp"/>

            <Button
android:id="@+id/btnTwo"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="2"
android:textSize="30dp"/>

            <Button
android:id="@+id/btnThree"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="3"
android:textSize="30dp"/>

 <Button
android:id="@+id/btnSub"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="-"
android:textSize="30dp"/>

        </LinearLayout>
        <LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

            <Button
android:id="@+id/btnZero"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="0"
android:textSize="30dp"/>

            <Button
android:id="@+id/btnDot"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="."
android:textSize="30dp"/>
            <Button
```

```
android:id="@+id/btnEqual"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="="
android:textSize="30dp"/>

            <Button
android:id="@+id/btnAdd"
android:layout_width="90dp"
android:layout_height="90dp"
android:text="+"
android:textSize="30dp"/>

        </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">

         <Button
android:id="@+id/btnClear"
android:layout_width="match_parent"
android:layout_height="50dp"
android:text="C"
android:textSize="20sp"/>

     </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

**Source Code:**

MainActivity.java

```
package ty.practical6;

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

import android.view.View;

import android.widget.Button;

 import android.widget.TextView;

 import java.text.DecimalFormat;

public class MainActivity extends AppCompatActivity

 {


    private double num1, num2, answer;     private char op;     private boolean
hasDot;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);


 final Button btnOne = (Button) findViewById(R.id.btnOne);

final Button btnTwo = (Button) findViewById(R.id.btnTwo);

final Button btnThree = (Button) findViewById(R.id.btnThree);

final Button btnFour = (Button) findViewById(R.id.btnFour);

final Button btnFive = (Button) findViewById(R.id.btnFive);

final Button btnSix = (Button) findViewById(R.id.btnSix);

final Button btnSeven = (Button) findViewById(R.id.btnSeven);

final Button btnEight = (Button) findViewById(R.id.btnEight);

final Button btnNine = (Button) findViewById(R.id.btnNine);

final Button btnZero = (Button) findViewById(R.id.btnZero);

final Button btnAdd = (Button) findViewById(R.id.btnAdd);
final Button btnSub = (Button) findViewById(R.id.btnSub);

final Button btnMultiply = (Button) findViewById(R.id.btnMultiply);

final Button btnDivide = (Button) findViewById(R.id.btnDivide);

final Button btnDot = (Button) findViewById(R.id.btnDot);

final Button btnEqual = (Button) findViewById(R.id.btnEqual);

final Button btnClear = (Button) findViewById(R.id.btnClear);

final TextView txtDisplay = (TextView) findViewById(R.id.txtDisplay);

        btnOne.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)

          {

            txtDisplay.append("1");

            }

        });

        btnTwo.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)

           {

               txtDisplay.append("2");

            }

        });

        btnThree.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)

{

             txtDisplay.append("3");

}

        });

        btnFour.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)

{
```

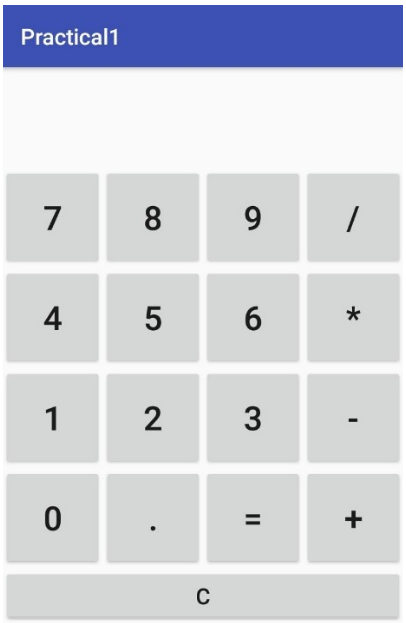
```
                 txtDisplay.append("4");
}
        });

        btnFive.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
            {
              txtDisplay.append("5");
            }
        });

        btnSix.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
           {
                txtDisplay.append("6");
            }
        });

        btnSeven.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
           {
                txtDisplay.append("7");
            }
        });

        btnEight.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
            {
                  txtDisplay.append("8");
            }
        });
        btnNine.setOnClickListener(new View.OnClickListener() {
public void  onClick(View v)

{

                         txtDisplay.append("9");

            }

        });


        btnZero.setOnClickListener(new View.OnClickListener() {
public void  onClick(View v)

                  {

                        txtDisplay.append("0");

            }

        });


        btnDot.setOnClickListener(new View.OnClickListener() {

public void  onClick(View v)

{

                //if Dot(.) is pressed then set hasDot to true to restrict
if(hasDot==false)
{
                        txtDisplay.append(".");
                        hasDot = true;
                }

            }

        });


        btnAdd.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v)

{

                num1 = Double.parseDouble(txtDisplay.getText().toString());

op = '+';

txtDisplay.setText("");
```

```
hasDot = false; //set hasDot to false to use in the next operand.
      }
    });

    btnSub.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
{
        num1 = Double.parseDouble(txtDisplay.getText().toString());
         op = '-';
        txtDisplay.setText("");
        hasDot = false; //set hasDot to false to use in the next operand.
      }
    });

    btnMultiply.setOnClickListener(new View.OnClickListener() {


public void onClick(View v)


{


num1 = Double.parseDouble(txtDisplay.getText().toString());


op = '*';


txtDisplay.setText("");


hasDot = false; //set hasDot to false to use in the next operand.
```

```
        }

      });

      btnDivide.setOnClickListener(new View.OnClickListener() {
public void onClick(View v){

            num1 = Double.parseDouble(txtDisplay.getText().toString());

             op = '/';

             txtDisplay.setText("");

            hasDot = false; //set hasDot to false to use in the next operand.

        }

      });

      btnEqual.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)

{

            num2 = Double.parseDouble(txtDisplay.getText().toString());

 switch (op)
{
case '+':
answer = num1 + num2;
break;
case '-':
answer = num1 - num2
break;
case '*':
answer = num1 * num2;
break;
case '/':
answer = num1 / num2;
break;
default:
break;
 }

DecimalFormat d = new DecimalFormat();

String ans = d.format(answer);
```

```
 txtDisplay.setText(ans);
hasDot = false; //set hasDot to false to use in new calculation.
            }
        });

        btnClear.setOnClickListener(new View.OnClickListener() {
public void onClick(View v)
{
txtDisplay.setText("");
hasDot = false; //set hasDot to false to use in new calculation.
            }
        });
    }
}
```

**Output:**



# Practical 7

# Develop an application for working with Menus and Screen Navigation.

**XML Code:**

**[Create a menu - Android Resource Directory and create a main_menu.xml – Android Resource File in it.]**

main_menu.xml

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

   <  ite
      m
      android:id="@+id/item1"
      android:title="FYCS" />
    <item
      android:id="@+id/item2"
      android:title="SYCS" />

   <   item
      android:id="@+id/item3"
      android:title="TYCS" />
</menu>
```

**Source Code:**

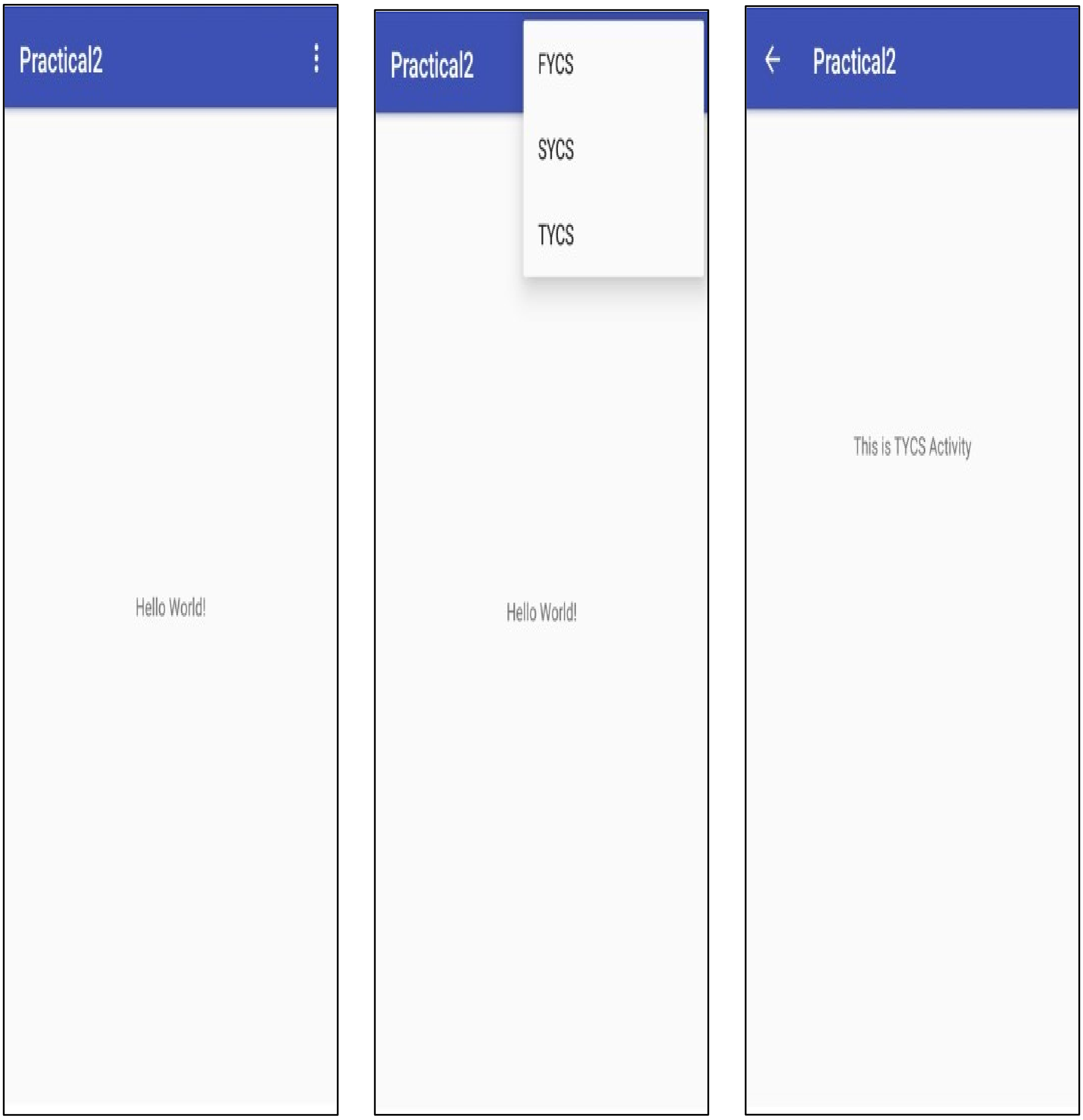
**[Using the MenuInflater link the main_menu.xml file in the MainActivity file]**

MainActivity.java

```
package ty.practical2;

 import android.content.Intent; import
android.os.Bundle; import
android.support.v7.app.AppCompatActivity; import
android.view.Menu; import
android.view.MenuInflater; import
android.view.MenuItem;

 import ty.practical7.R;

 public class MainActivity extends AppCompatActivity {
   @Override     protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

   }



        @Override public boolean onCreateOptionsMenu(Menu
        menu) {

           MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main_menu, menu);


           return true;
        }

         @Override public boolean onOptionsItemSelected(MenuItem
        item) {

            switch(item.getItemId())

           {        case R.id.item1:            s

        startActivity(new Intent(MainActivity.this, FYCS.class));

        return true;

        case R.id.item2:

        startActivity(new Intent(MainActivity.this, SYCS.class));

        return true;

        case R.id.item3:

        startActivity(new Intent(MainActivity.this, TYCS.class));

         return true;

        default:

                 return super.onOptionsItemSelected(item);

           }

        }

}
```

**[Create 3 new activities to open when the menu items in the options menu is clicked/selected named as FYCS, SYCS, TYCS]**

**Output:**



Hello World!

Hello World!

This is TYCS Activity

# Practical 8

## Develop an application for working with notifications.

### activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context="MainActivity">
    <Button
        android:id="@+id/btnCreate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="199dp"
        android:onClick="CreateNotification"
        android:text="Create Notification" />
</RelativeLayout>
```

### activity_dummy.xml

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

<RelativeLayout 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"

    tools:context="DummyActivity">

    <TextView

        android:id="@+id/textView"

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

       android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="207dp"

        android:text="This is Dummy Activity" />

</RelativeLayout>
```

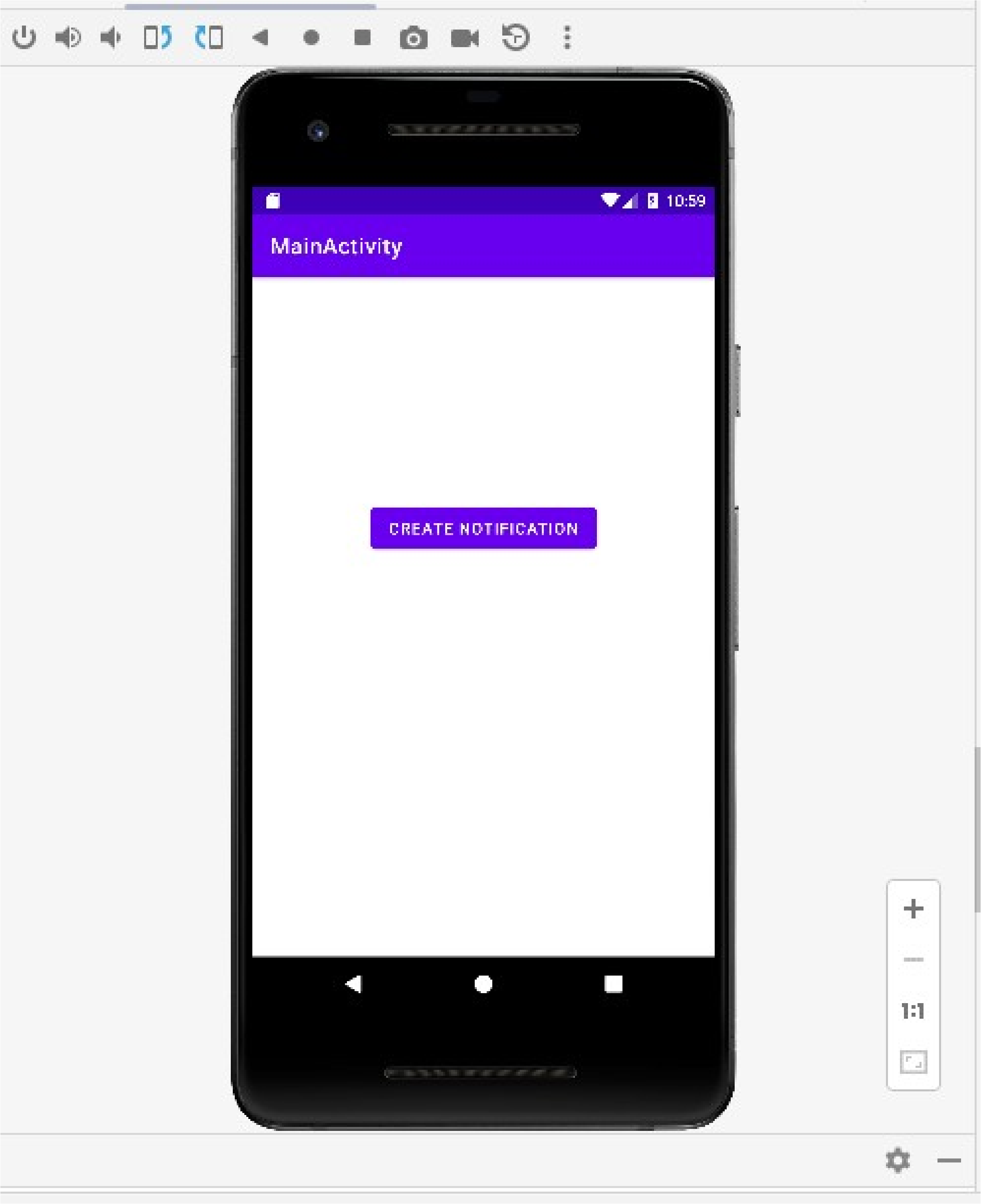
## MainActivity.java

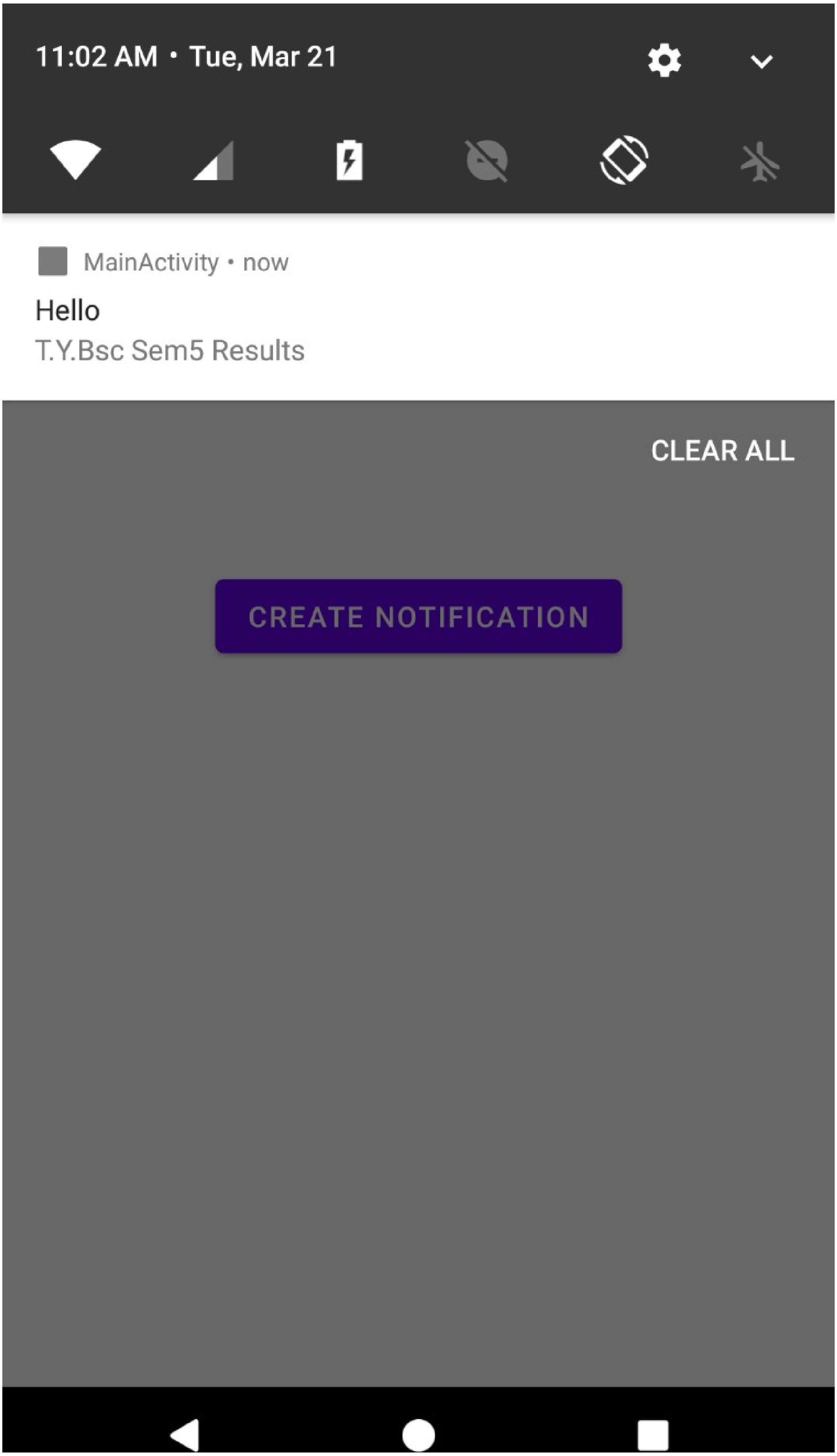
```
package com.example.mainactivity;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity
{

    @Override

    protected void onCreate(Bundle savedInstanceState)
{

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }

    int notifyID = 1;

    int numMessages = 0;

    public void CreateNotification(View v) {

        numMessages+=1;

        NotificationManager notificationManager = (NotificationManager)

                getSystemService(NOTIFICATION_SERVICE);

        Intent intent = new Intent(this, DummyActivity.class);
        // use System.currentTimeMillis() to have a unique ID for the pending
        PendingIntent pIntent = PendingIntent.getActivity(this, (int)
              System.currentTimeMillis(), intent, 0);
        NotificationCompat.Builder n = new NotificationCompat.Builder(this)
                .setContentTitle("Hello Dear Studens")
                .setContentText("T.Y.Bsc Sem5 Results")
                .setContentIntent(pIntent)
                .setSmallIcon(R.mipmap.ic_launcher)
                .setNumber(numMessages)
                .setAutoCancel(true);
        notificationManager.notify(notifyID, n.build());
    }
}
```

## DummyActivity.java

```
package com.example.mainactivity;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class DummyActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dummy);
    }
}
```

**OUTPUT**


10:59
MainActivity
CREATE NOTIFICATION
1:1



MainActivity • now

Hello

T.Y.Bsc Sem5 Results

# Practical 9

# Design a simple to-do list application using SQLite

**XML Code:**

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context="ty.practical5.MainActivity">
    <ListView
        android:id="@+id/lvData"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/btnAdd"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true" />
    <EditText
        android:id="@+id/txtItem"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignTop="@+id/btnAdd"
android:layout_toLeftOf="@+id/btnAdd"
android:layout_toStartOf="@+id/btnAdd"
android:hint="Enter a New Item"
android:inputType="textMultiLine"
        />

    <Button
        android:id="@+id/btnAdd"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:onClick="AddItem"
android:text="Add Item" />
</RelativeLayout>
```

activity_task_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout    andro="http://schemas.android.com/apk/res/
xmlns:             id    android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
     andro:layout_width="match_parent"
     id
     andro:layout_height="match_parent"
     id    context="ty.practical5.TaskDetails">
     tools:

    <Button
        android:id="@+id/btnUpdate"
           andro:layout_width="wrap_content"
           id
           andro:layout_height="wrap_content"
           id
           andro:layout_alignParentStart="true"
           id
           andro:layout_below="@+id/txtData"
           id
           andro:layout_marginStart="48dp"
           id
           andro:layout_marginTop="50dp"
           id
           andro:onClick="Update"
           id
           andro:text="Update" />
           id


    <Button
        android:id="@+id/btnDelete"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignEnd="@+id/txtData"
android:layout_alignTop="@+id/btnUpdate"
android:layout_marginEnd="13dp"
android:onClick="Delete"
android:text="Delete" />

    <EditText
           andro:id="@+id/txtData"
           id
           andro:layout_width="wrap_content"
           id
           andro:layout_height="wrap_content"
           id
        andro :layout_alignParentTop="true"
        id
        andro :layout_alignStart="@+id/btnUpdate"
        id
        andro :layout_marginStart="19dp"
        id
        andro :layout_marginTop="57dp"
        id
        andro :ems="10"
        id
        andro :inputType="textMultiLine" />
        id

</RelativeLayout>
```

## Source Code:

toDoDatabaseHelper.java

```
package ty.practical9;

import
android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;


public class toDoDatabaseHelper extends SQLiteOpenHelper {


   private static final String DATABASE_NAME = "todoList.db";
private static final int DATABASE_VERSION = 1;

   public toDoDatabaseHelper(Context context) {

      super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }


   @Override
   public void onCreate(SQLiteDatabase db)
{        String query = "CREATE TABLE ToDo (task
TEXT)";        db.execSQL(query);

   }


   @Override
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {        db.execSQL("DROP TABLE IF EXISTS ToDo");

onCreate(db);     }


   public void addTask(String item){

      ContentValues values = new ContentValues();

     values.put("task", item);

      SQLiteDatabase db = getWritableDatabase();
db.insert("ToDo", null, values);
```

```
 db.close();

    }


    //Delete a product from the database
public void deleteTask(String item){
        SQLiteDatabase db = this.getWritableDatabase();

db.execSQL("DELETE FROM ToDo where task='"+item+"'");

    }


    public void updateTask(String oldvalue, String newvalue){


        try {
            SQLiteDatabase db = this.getWritableDatabase();

 ContentValues contentValues = new ContentValues();

 contentValues.put("task", newvalue);

            db.update("ToDo", contentValues, "task='"+oldvalue +"'",
null);

db.close();

} catch (Exception e) {

            e.printStackTrace();

        }


    }


    public ArrayList<String> getAllTasks() {


        ArrayList<String> contactList = new ArrayList<String>();


        String selectQuery = "SELECT  * FROM
ToDo";

        SQLiteDatabase db = this.getWritableDatabase();

        Cursor cursor = db.rawQuery(selectQuery, null);
        if (cursor.moveToFirst()) {
do {

                contactList.add(cursor.getString(0));

            } while (cursor.moveToNext());

        }

        return contactList;

    }


    public ArrayList<String> getTaskByItem(int item) {


        ArrayList<String> contactList = new ArrayList<String>();

        String selectQuery = "SELECT  * FROM ToDo where task="
+item;

        SQLiteDatabase db = this.getWritableDatabase();

        Cursor cursor = db.rawQuery(selectQuery, null);


        if (cursor.moveToFirst()) {

            contactList.add(cursor.getString(1));

        }

        return contactList;

    }

}
```

MainActivity.java

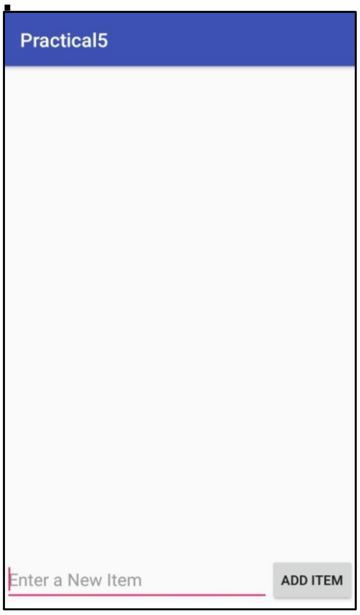
```
package ty.practical9;

import android.content.Intent;
import android.os.Bundle;

import
android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.AdapterView; import
android.widget.ArrayAdapter; import
android.widget.EditText; import
android.widget.ListView;


import java.util.ArrayList;


public class MainActivity extends AppCompatActivity {


    private ArrayList<String> items;
private ArrayAdapter<String>
itemsAdapter;     private ListView lvData;

    private toDoDatabaseHelper dbAccess;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);


        dbAccess = new toDoDatabaseHelper(this);
lvData = (ListView) findViewById(R.id.lvData);

items = new ArrayList<String>();

readItems();


        itemsAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, items);
lvData.setAdapter(itemsAdapter);


        lvData.setOnItemClickListener(new AdapterView.OnItemClickListener() {

            @Override
            public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {

                Intent intent = new Intent(MainActivity.this, TaskDetails.class);
intent.putExtra("data", lvData.getItemAtPosition(position).toString());
startActivity(intent);

                // Refresh the adapter
refreshListView();
            }

        });

    }


    public void AddItem(View v) {

        EditText txtItem = (EditText) findViewById(R.id.txtItem);
String itemText = txtItem.getText().toString();
itemsAdapter.add(itemText);         txtItem.setText("");

dbAccess.addTask(itemText);

    }


    public void readItems() {
try {
            items = new ArrayList<String>(dbAccess.getAllTasks());

        } catch (Exception e) {

            items = new ArrayList<String>();

        }

    }


    public void refreshListView() {


        itemsAdapter.notifyDataSetChanged();

    }

}
```

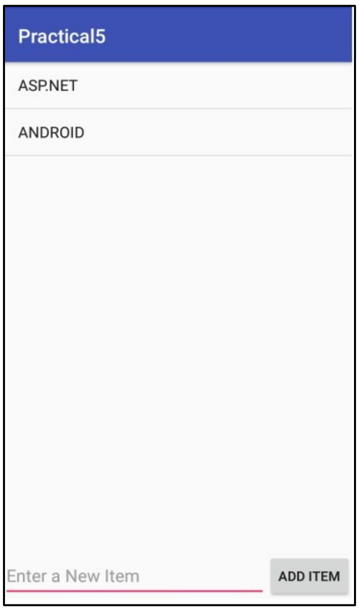
TaskDetails.java

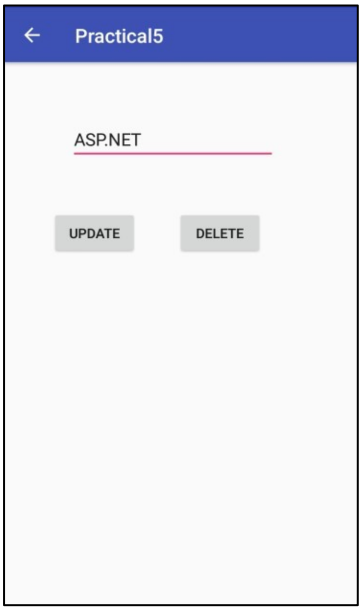
```
package ty.practical9;

import android.content.Intent;

import
android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import
android.view.MenuItem; import
android.view.View; import
android.widget.EditText;

public class TaskDetails extends AppCompatActivity {

    private toDoDatabaseHelper dbAccess;

    String oldvalue="";

    EditText txtData;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_task_details);

        //code to enable the back button
        getSupportActionBar().setDisplayShowHomeEnabled(true);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);


        //code to fetch the selected list item data in the previous activity
dbAccess = new toDoDatabaseHelper(this);

        Intent intent = getIntent();

        oldvalue = intent.getStringExtra("data");

        txtData = (EditText) findViewById(R.id.txtData);
txtData.setText(intent.getStringExtra("data"));

    }

    //code for delete button to delete the task
protected void Delete(View v) {
        dbAccess.deleteTask(txtData.getText().toString());

        Intent intent = new Intent(TaskDetails.this,MainActivity.class) ;
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(intent);        finish();
```

```
    }


    //code for delete button to update the task
protected void Update(View v) {
        dbAccess.updateTask(oldvalue, txtData.getText().toString());
Intent intent = new Intent(TaskDetails.this,MainActivity.class) ;
intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(intent);        finish();

    }


    //code to close the current activity and move to the previous
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
finish();

        return super.onOptionsItemSelected(item);

    }

}
```

**Output**







# Practical 10

## Programming on Intents , Events , Listeners and Adapters Demonstrating The Android Intent Class

## Introduction :

An intent is a messaging object used to request any action from another app component. Intents facilitate communication between different components in several ways. The intent is used to launch an activity, start the services, broadcast receivers, display a web page, dial a phone call, send messages from one activity to another activity, and so on.

There are two types of intents in android

1) Implicit Intent

2) Explicit Intent

## Create an android application to pass the data from one activity to another activity in the same application using intent

In this example we have two activities, activity_first which are the source activity, and activity_second which is the destination activity. We can send the data using the putExtra() method from one activity and get the data from the second activity using the getStringExtra() method.

- **putExtra()** method is used for sending the data, data in key-value pair key is variable name and value can be Int, String, Float, etc.
- **getStringExtra()** method is for getting the data(key) that is sent by the above method.

According to the data type of value, there are other methods like getIntExtra(), getFloatExtra()

**Step 1 : Create New Project Go to File → New→ New Project → select Empty activity → Next → Set Name for your project as IntentDemo and click on Finish**

## Main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
   tools:context=".MainActivity">
   <Button
      android:id="@+id/btn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Button"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Step 3 : Create another activity(Right click on app → New → Activity → Empty Activity → Set activity name as SecondActivity and click on finish.)

## Step 4 : Add following code activity_second.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
   tools:context=".SecondActivity">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World Welcome to model college"
      android:textSize="30sp">
   </TextView>
</LinearLayout>
```

# Step 5 : Add following code

## MainActivity.java

```
package com.example.mainactivity;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity
{
Button button;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
```

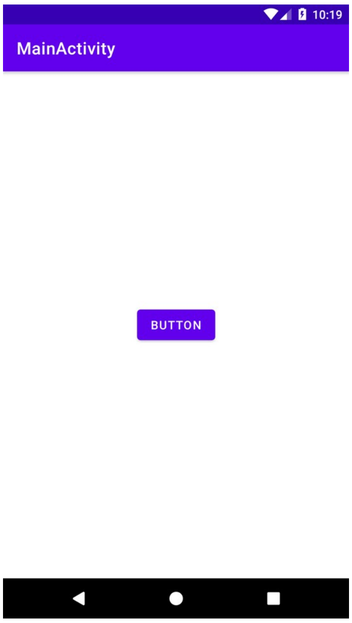
```
        setContentView(R.layout.activity_main);

        button=findViewById(R.id.btn);
        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                Intent intent=new Intent(MainActivity.this, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

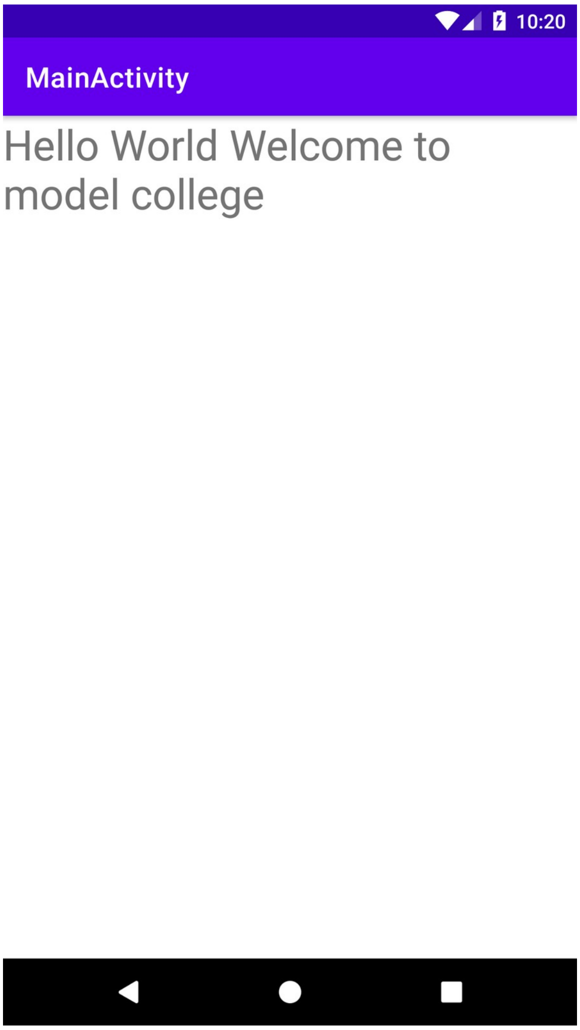
## Step 6 : Add following code

## SecondActivity.java

```
package com.example.mainactivity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class SecondActivity extends AppCompatActivity {

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

OUTPUT





Hello World Welcome to model college

# Demonstrating Events and Event Listeners

## Introduction

Events are represented as instances of the **Event class or its subclasses**. An event is an object that encapsulates information about something that has happened, such as a user clicking a button or a network request completing.

To handle events, we can use **event listeners**. An event listener is a function or lambda that gets executed when an event occurs. Event listeners can be attached to an object that can produce events, such as a button or a text field.

Any View (Button, TextView, etc) has many event listeners that can be attached using the setOnEvent pattern which involves passing a class that implements a particular event interface. The listeners available to any View include:

- **setOnClickListener -** Callback when the view is clicked
- **setOnKeyListener -** Callback for pressing a hardware key when view has focus
- **setOnLongClickListener -** Callback for pressing and holding a view
- **setOnTouchListener -** Callback for touching down or up on a view. Used to detect any form of contact with the touch screen including individual or multiple touches and gesture motions.
- **setOnDragListener** - Callback when the view is dragged
- **setOnFocusChangeListener** - Callback when the view changes focus
- **setOnGenericMotionListener** - Callback for arbitrary gestures
- **setOnHoverListener -** Callback for hovering over the view

## Step 1 : Create New Project

Go to File → New→ New Project → select Empty activity → Next → Set Name for your project as EventListenerDemo and click on Finish

## Step 2 : Add following code in

### activity_main.xml

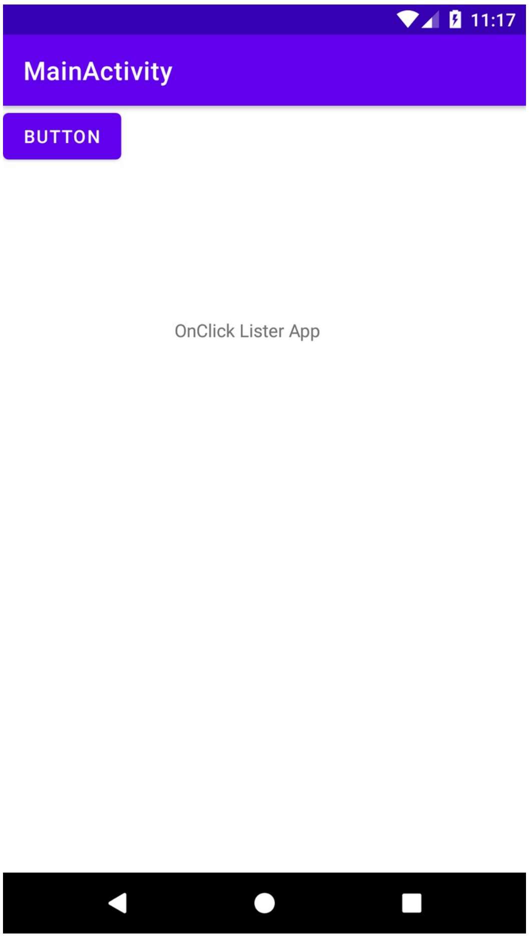
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="OnClick Lister App"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.455"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.286" />

    <Button
        android:id="@+id/btnOK"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        tools:layout_editor_absoluteX="156dp"
        tools:layout_editor_absoluteY="548dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

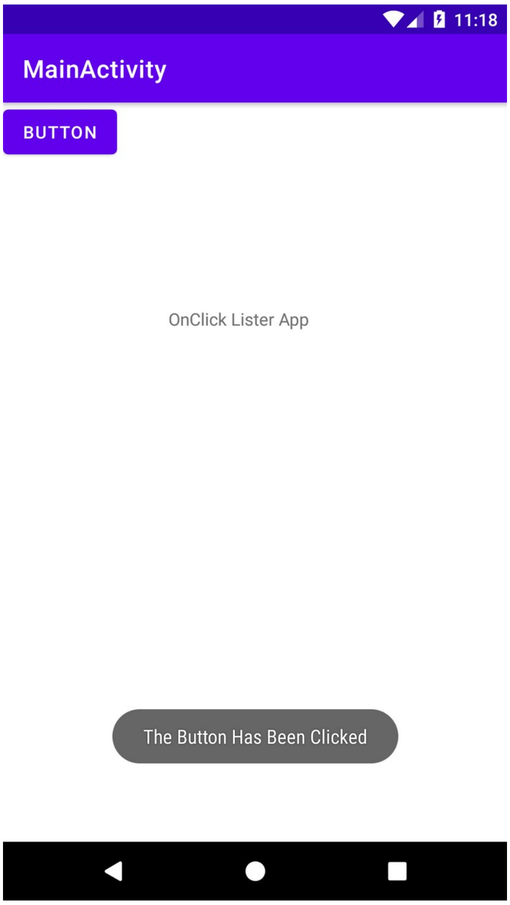
Step 3 : Add following code

**MainActivity.java**

```
package com.example.mainactivity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Button btn=findViewById(R.id.btnOK);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view)
            {
                Toast.makeText(getApplicationContext(), "The Button Has Been
Clicked",Toast.LENGTH_LONG).show();

            }
        });
    }
}
```

# OUTPUT





The Button Has Been Clicked

**Create an android application, with one button and image view also set the background image in the application ,the image should be changed on each click of the button (Use two images only)**

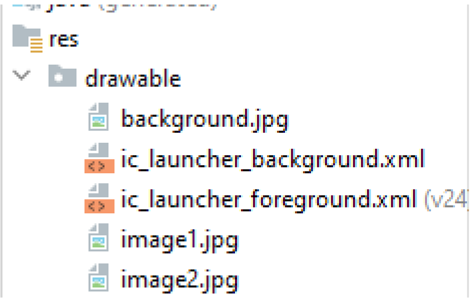
Step 1 : Create New Project Go to File → New→ New Project → select Empty activity → Next → Set Name for your project as ImageChangeDemo and click on Finish

Step 2 : Add following code in

### activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
   tools:context=".MainActivity">
   <ImageButton
      android:id="@+id/imageButton"
      android:layout_width="106dp"
      android:layout_height="106dp"
      android:layout_marginTop="50dp"
      app:layout_constraintBaseline_toEndOf="parent"
      app:layout_constraintBaseline_toStartOf="parent"
      app:layout_constraintBaseline_toTopOf="parent"
      android:src="@drawable/image2"
      android:contentDescription="TODO" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## Step 3 : Copy any three images in res/drawable folder



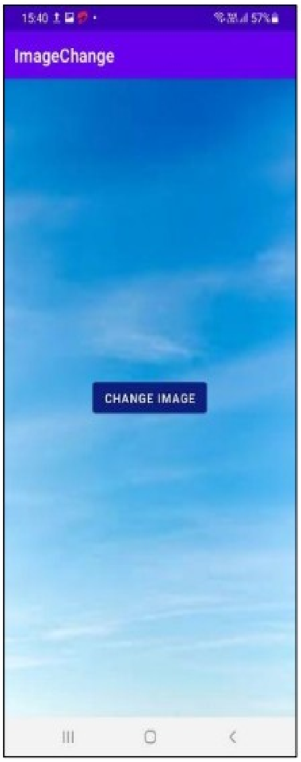
res
drawable
background.jpg
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
image1.jpg
image2.jpg

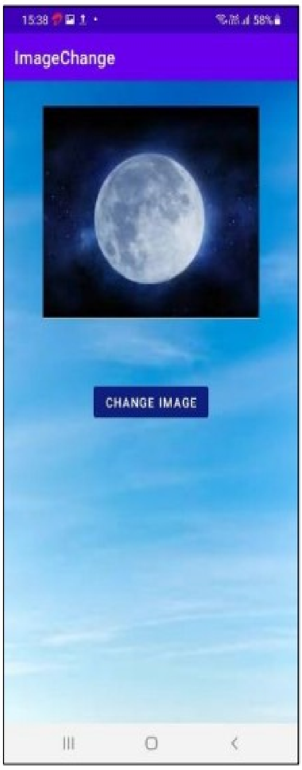
# Step 4 : Add following code

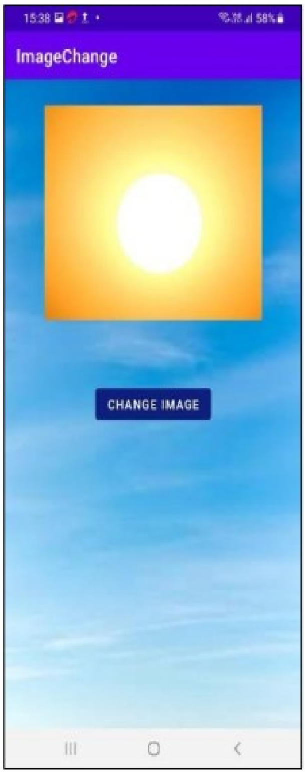
## MainActivity.java file

```
package com.example.manactivity;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity
{
    ImageButton imgButton;
    @Override
    protected void onCreate(Bundle SavedInstanceState)
    {
        super.onCreate(SavedInstanceState);
        setContentView(R.layout.activity_main);

        imgButton=(ImageButton)findViewById(R.id.imageButton);
        imgButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(MainActivity.this, "You Click image Button",
Toast.LENGTH_SHORT).show();
            }
        });

    }
}
```

# OUTPUT


15:40
57%
ImageChange
CHANGE IMAGE


15:38
58%
ImageChange
CHANGE IMAGE


15:38
58%
ImageChange
CHANGE IMAGE