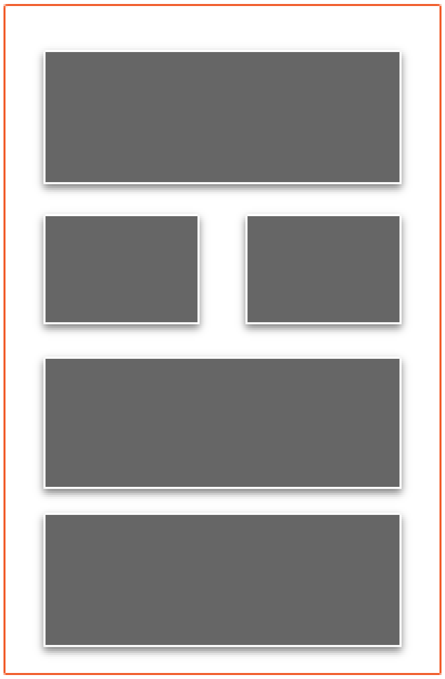
Lab 4: Android RelativeLayout

Introduction

Relative Layout Specify the positions of views relative to other views.



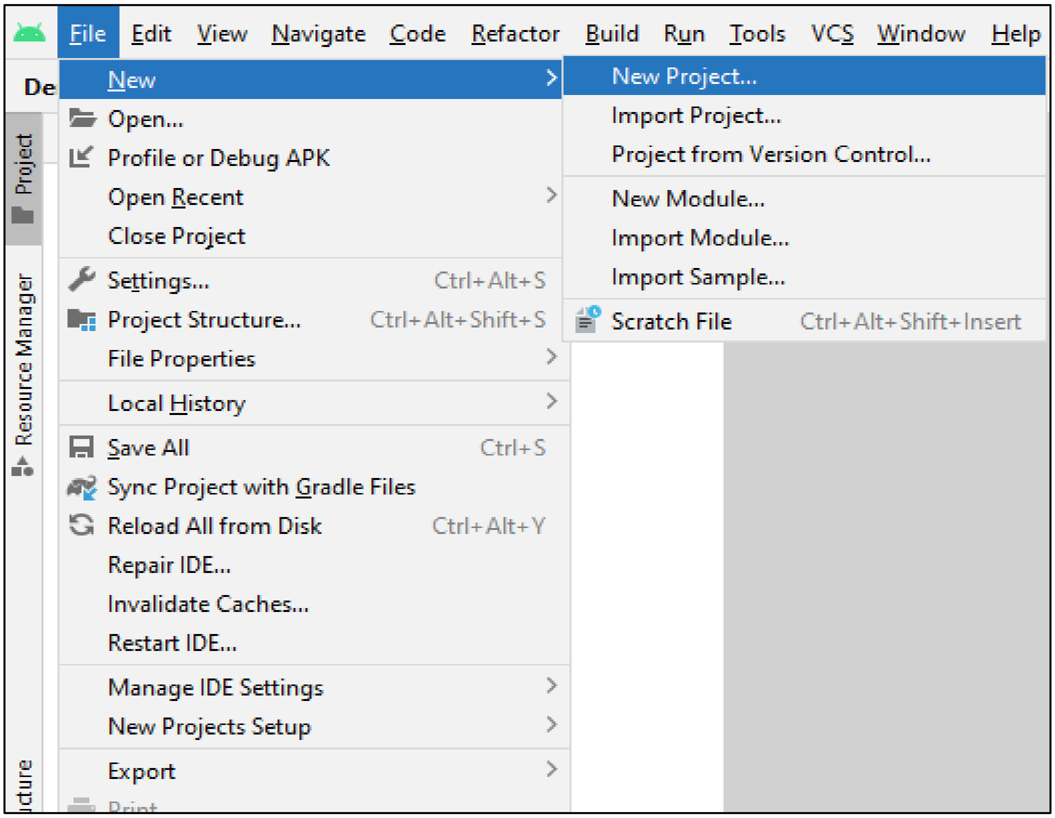
Relative Layout Attributes

Attribute	Description
id	Used to uniquely specify
gravity	Used to specify child position
ignoreGravity	Used to specify which need to be ignored for gravity

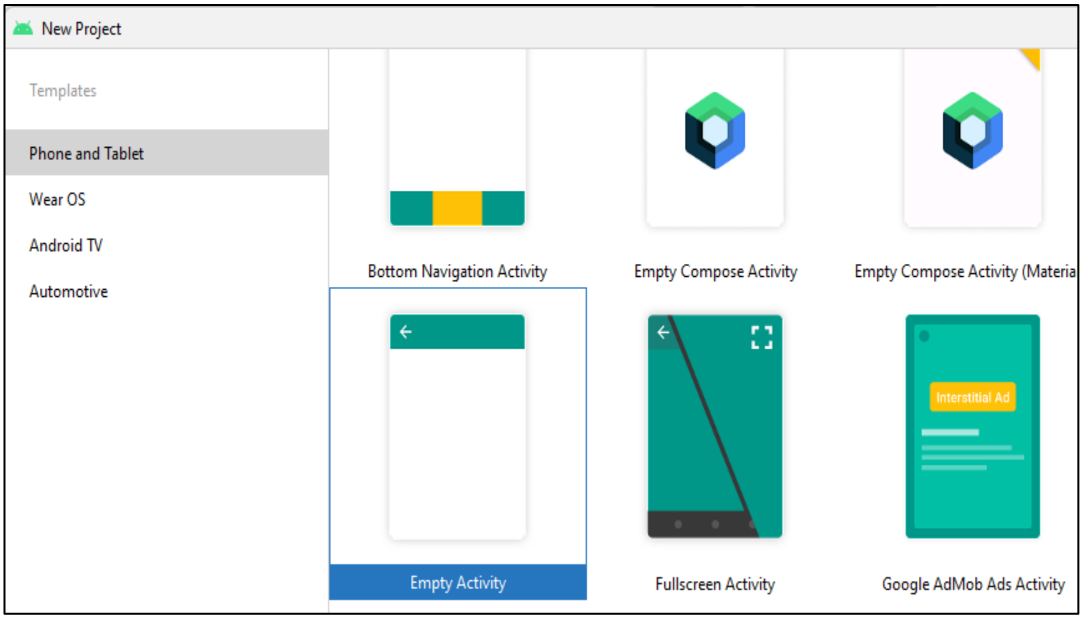
Let’s get Started

You'll be guided through easy stages in this exercise to design your own Android application utilising Relative Layout.

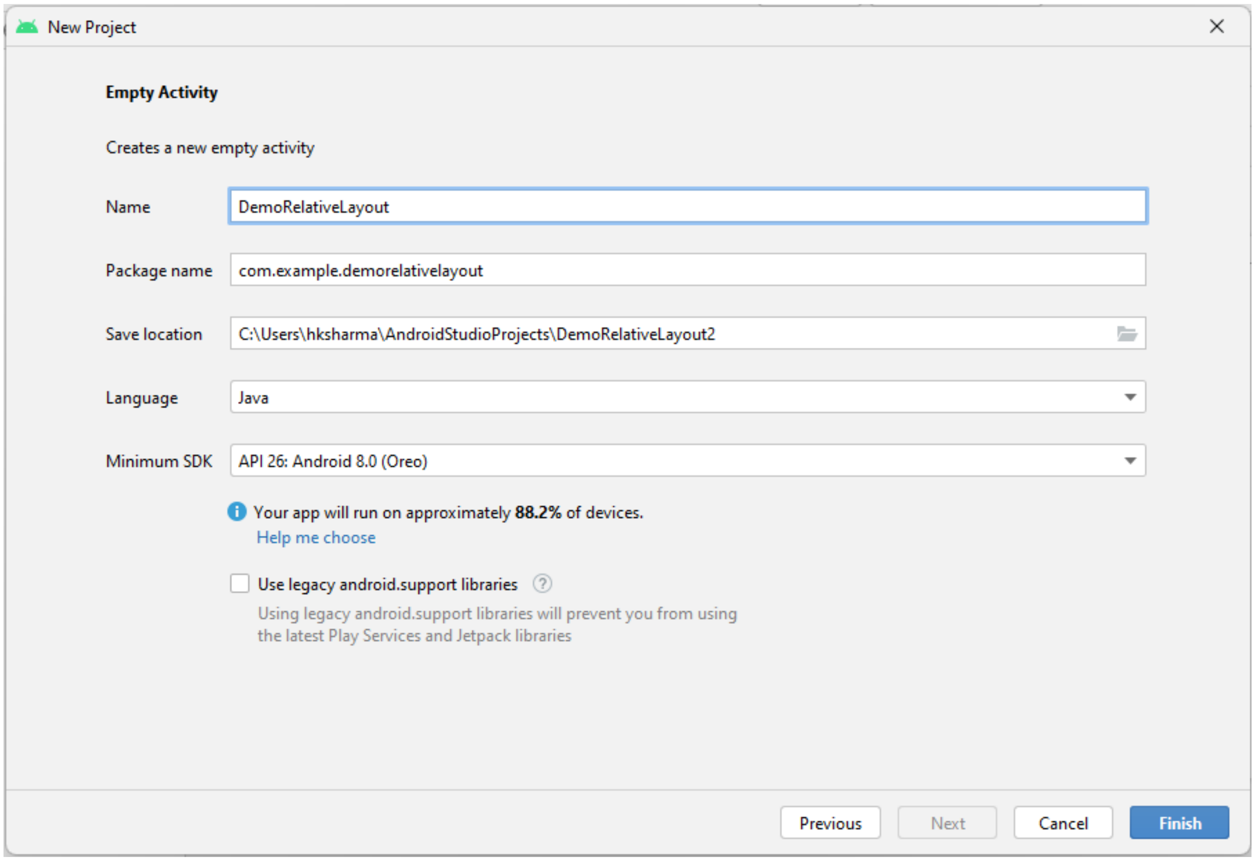
Step 1: Create a New Project in Android Studio as shown below



Step 2: Select Empty Activity as shown below



Step 3: Provide a Project Name as shown below



Empty Activity

Creates a new empty activity

Name: DemoRelativeLayout

Package name: com.example.demorelativelayout

Save location: C:\Users\hksharma\AndroidStudioProjects\DemoRelativeLayout2

Language: Java

Minimum SDK: API 26: Android 8.0 (Oreo)

Your app will run on approximately **88.2%** of devices.

Help me choose

□ Use legacy android.support libraries

Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

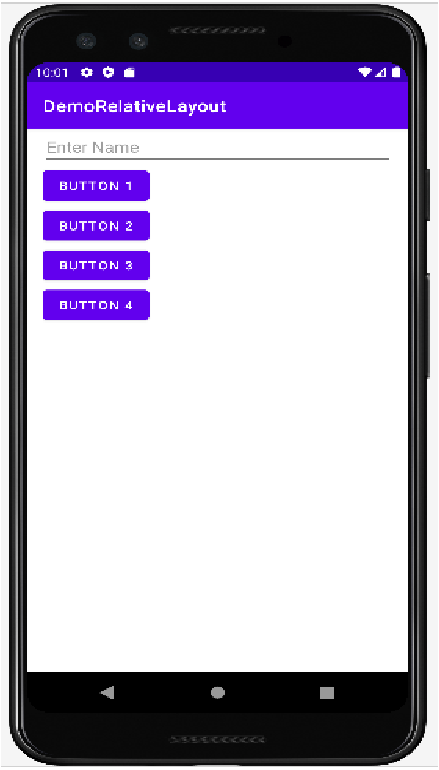
Step 4: Update MainActivity.java as per the code given below

```java
package com.example.demorelativelayout;
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);
    }
}
```

Step 5: Update activity_main.xml for Relative Layout as per the code given below

```
<?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/name" />

    <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="Button 1"
            android:id="@+id/button" />

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

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 3"
            android:id="@+id/button3" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button 4"
            android:id="@+id/button4" />

    </LinearLayout>

</RelativeLayout>
```

Step 6: Check Output on Android Emulator and it should look like as given below



Voila!! We have successfully completed this lab.