Mobile Application Development

Course Code:18CSL67

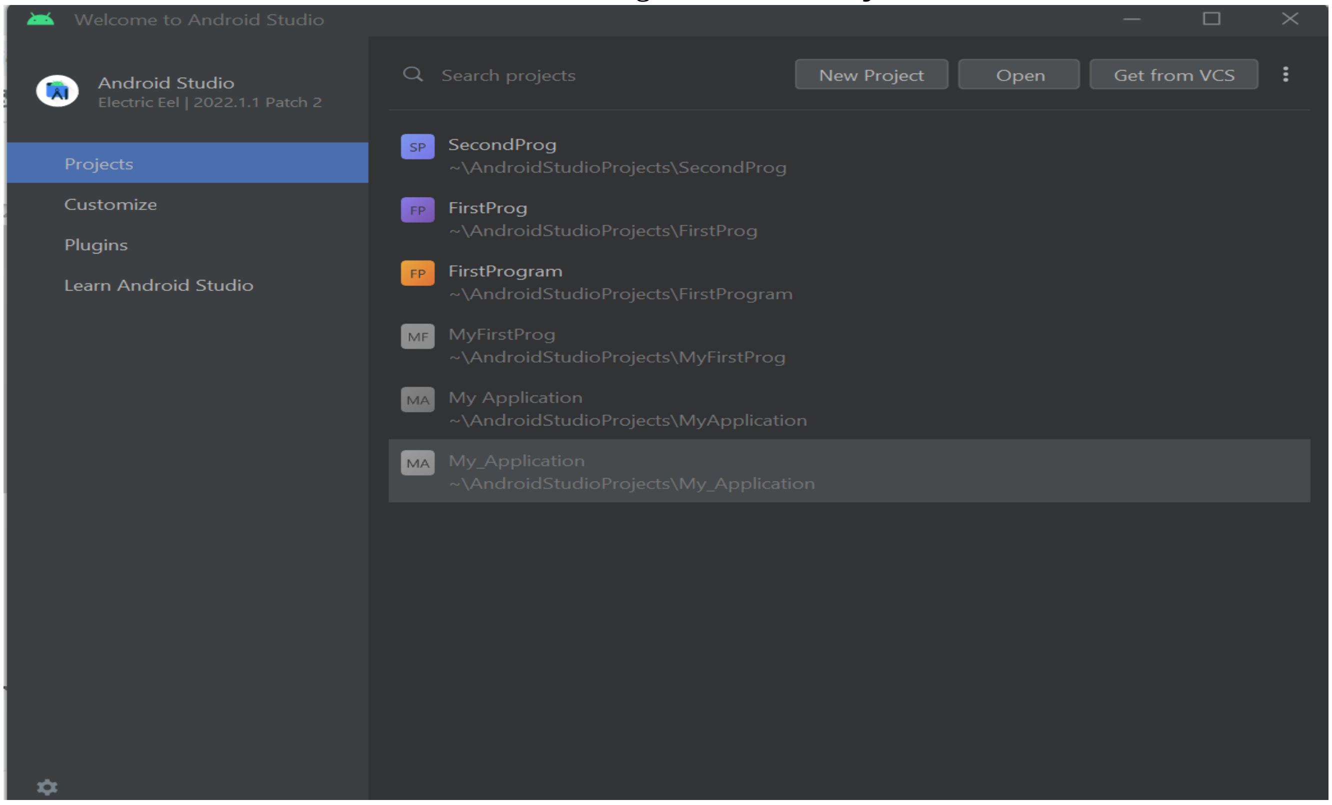
Part – A

Q1. In this lab we will be learning how to use and extend the Android user interface library.
a. Views, View Groups, Layouts, and Widgets are and how they relate to each other.
b. How to declare and reference resources in code.
c. How to navigate between multiple activities.
d. How to share the data between the activities.
e. Explore life-cycle methods of an activity.
f. How to use Events and Event Listeners.
g. How to create Toast Notifications.

Create your first project

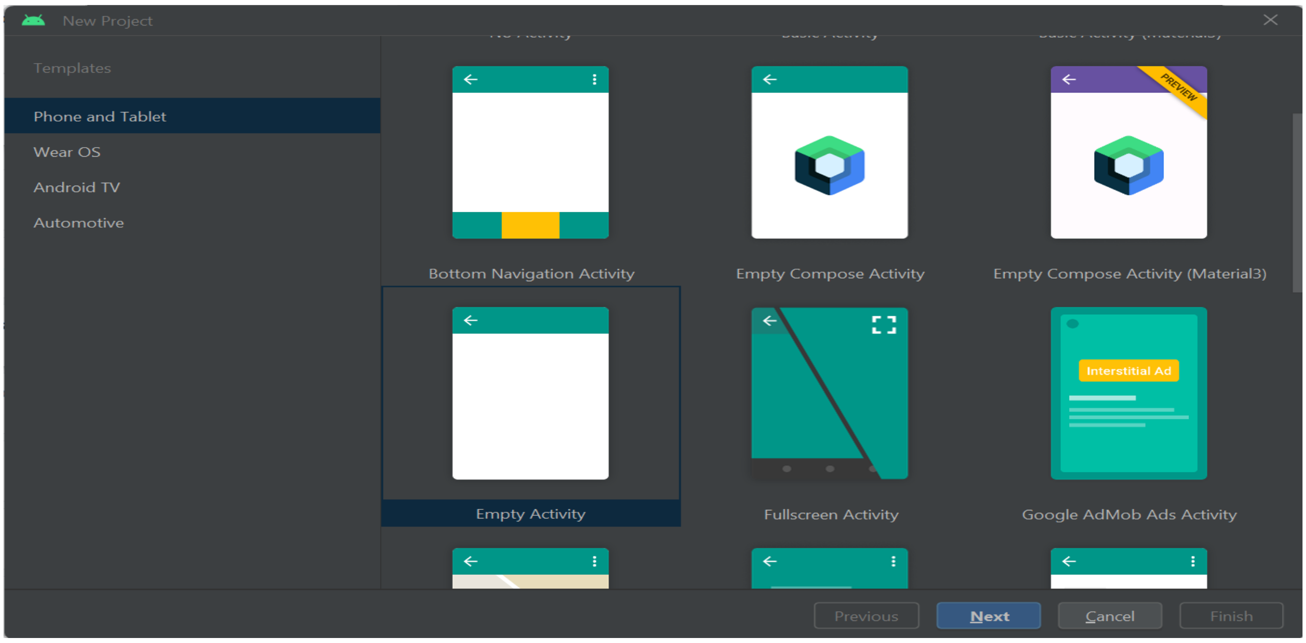
Step 1: Create a new project

1. Open Android Studio.
2. In the **Welcome to Android Studio** dialog, click **New Project**.

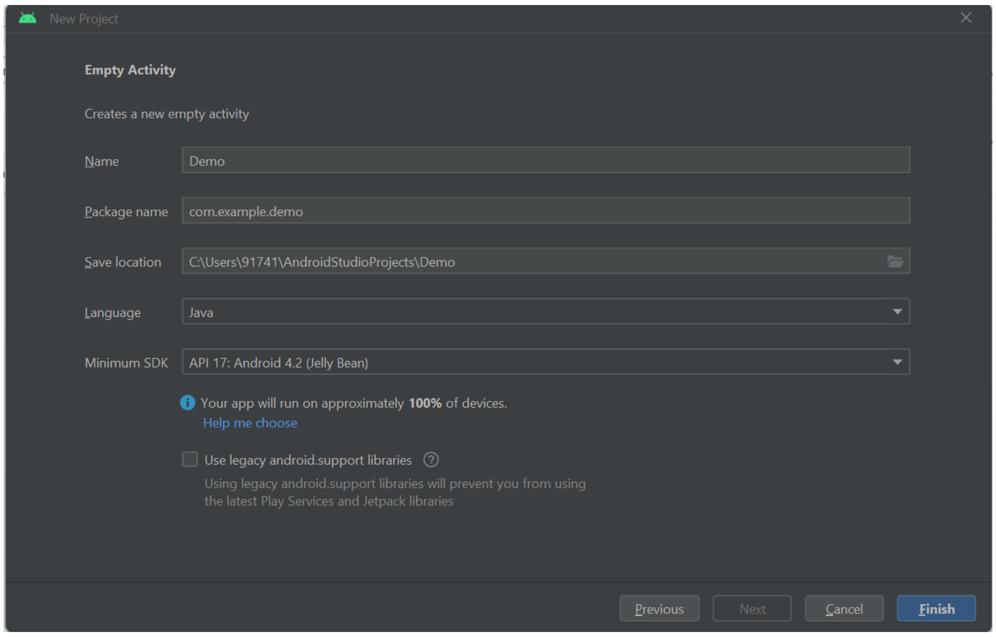


SecondProg
~\AndroidStudioProjects\SecondProg
FirstProg
~\AndroidStudioProjects\FirstProg
FirstProgram
~\AndroidStudioProjects\FirstProgram
MyFirstProg
~\AndroidStudioProjects\MyFirstProg
My Application
~\AndroidStudioProjects\MyApplication
My_Application
~\AndroidStudioProjects\My_Application

3.Select **Empty Activity** (not the default). Click **Next**.



4. Give your application a name such as **Demo**


New Project

Empty Activity

Creates a new empty activity

Name: Demo

Package name: com.example.demo

Save location: C:\Users\91741\AndroidStudioProjects\Demo

Language: Java

Minimum SDK: API 17: Android 4.2 (Jelly Bean)

Your app will run on approximately **100%** 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

5. Make sure the **Language** is set to **Java .**
6. Leave the defaults for the other fields.
7. Click **Finish**.

After these steps, Android Studio:

- Creates a folder for your Android Studio project called **MyFirstApp**. This is usually in a folder called **AndroidStudioProjects** below your home directory.

- Builds your project (this may take a few moments). Android Studio uses Gradle as its build system. You can follow the build progress at the bottom of the Android Studio window.
- Opens the code editor showing your project.

Code:

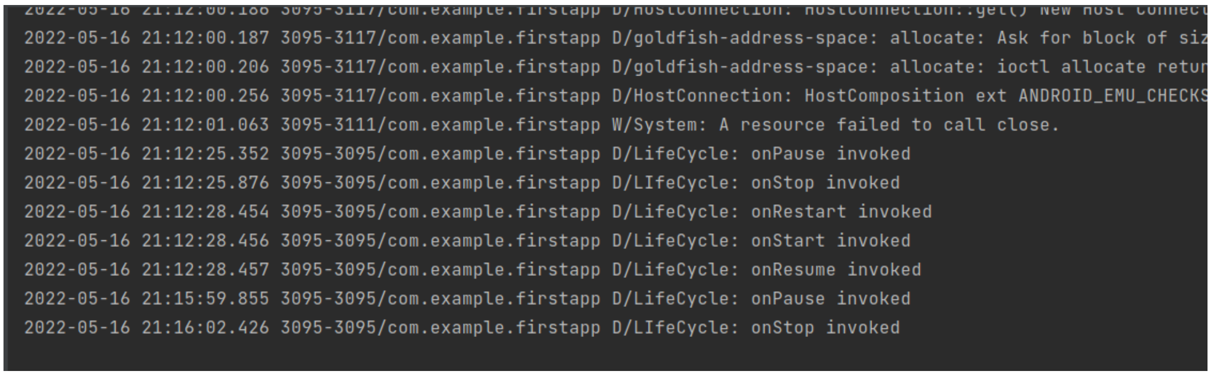
activity_main.xml

```
<?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="com.example.firstprog.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="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" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

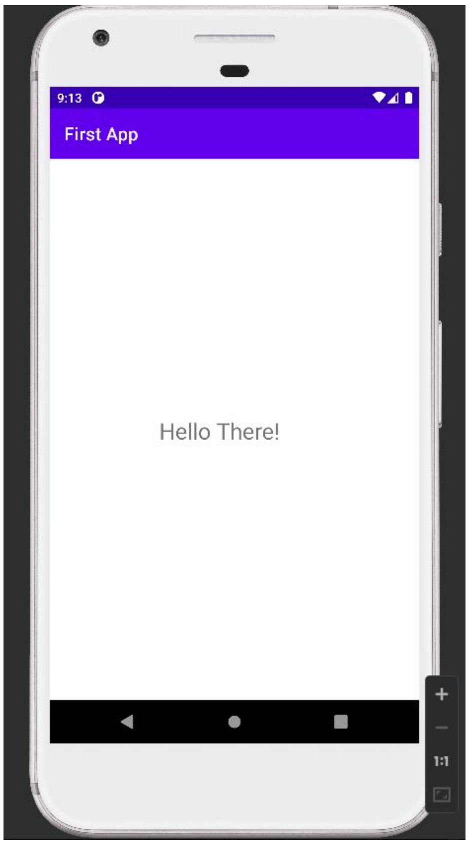
MainActivity.java

```
package com.example.firstprog;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle","onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle","onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle","onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle","onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("lifecycle","onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle","onDestroy invoked");
    }

}
```

<u>Output</u>:



```
2022-05-16 21:12:00.186 3095-3117/com.example.firstapp D/HostConnection: HostConnection::get() New Host Connect
2022-05-16 21:12:00.187 3095-3117/com.example.firstapp D/goldfish-address-space: allocate: Ask for block of siz
2022-05-16 21:12:00.206 3095-3117/com.example.firstapp D/goldfish-address-space: allocate: ioctl allocate retur
2022-05-16 21:12:00.256 3095-3117/com.example.firstapp D/HostConnection: HostComposition ext ANDROID_EMU_CHECKS
2022-05-16 21:12:01.063 3095-3111/com.example.firstapp W/System: A resource failed to call close.
2022-05-16 21:12:25.352 3095-3095/com.example.firstapp D/LifeCycle: onPause invoked
2022-05-16 21:12:25.876 3095-3095/com.example.firstapp D/LIfeCycle: onStop invoked
2022-05-16 21:12:28.454 3095-3095/com.example.firstapp D/LifeCycle: onRestart invoked
2022-05-16 21:12:28.456 3095-3095/com.example.firstapp D/LifeCycle: onStart invoked
2022-05-16 21:12:28.457 3095-3095/com.example.firstapp D/LifeCycle: onResume invoked
2022-05-16 21:15:59.855 3095-3095/com.example.firstapp D/LifeCycle: onPause invoked
2022-05-16 21:16:02.426 3095-3095/com.example.firstapp D/LIfeCycle: onStop invoked
```



Enable USB debugging

To let Android Studio communicate with your Android device, you must enable USB debugging in the Developer options settings of the device.

To show developer options and enable USB debugging:

1. On your Android device, tap **Settings** > **About phone**.
2. Tap **Build number** seven times.
3. If prompted, enter your device password or pin. You know you succeeded when you see a **You are now a developer!** message.
4. Return to Settings and then tap **System** > **Developer options**.
5. If you don't see **Developer options**, tap **Advanced options**.

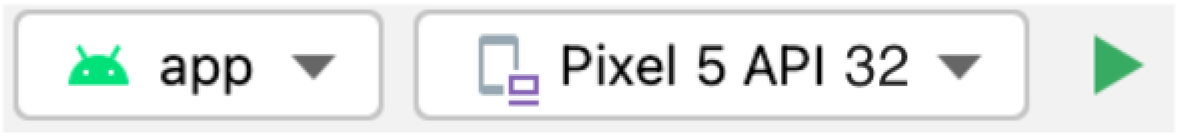
Run your app on the Android device with a cable

There are two ways to connect your device to Android Studio, through a cable or through Wi-Fi. You can choose whichever you like more.

To run your app from Android Studio on your Android device:

1. Connect your Android device to your computer with a USB cable. A dialog should appear on your device, which asks you to allow USB debugging.

2. Select the **Always allow from this computer** checkbox and then tap **OK**.
3. In Android Studio on your computer, make sure your device is selected in the dropdown. Click ▶.



4. Select your device and then click **OK**. Android Studio installs the app on your device and runs it.
5. If your device runs an Android platform that isn't installed in Android Studio and you see a message that asks whether you want to install the needed platform, click **Install > Continue > Finish**. Android Studio installs the app on your device and runs it.

Q2. You will expand on your knowledge of the Android user interface library.

a. How to declare layouts statically as an xml resource.

b. How to create custom Views from scratch to suit a specific need.

c. How to create Options and Context Menus.

d. How to use ListAdapter and ArrayAdapter to bind data source to a List View.

e. How to create AlertDialog and progress Dialog in your activity.

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"
    tools:visibility="visible">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Long Click on Me"
        android:textSize="25sp"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />


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

MainActivity.java

```
package com.example.secondprog;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    private Button button;

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

       button = findViewById(R.id.button);
        registerForContextMenu(button);
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        getMenuInflater().inflate(R.menu.menu,menu);
        super.onCreateContextMenu(menu, v, menuInfo);
    }

    @Override
    public boolean onContextItemSelected(@NonNull MenuItem item) {
        Toast.makeText(this, ""+item.getTitle(), Toast.LENGTH_SHORT).show();
        return super.onContextItemSelected(item);
    }
}
```

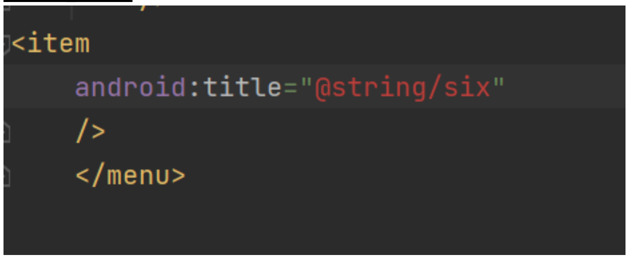
Note: on the ***menu*** right click and create a menu.xml

menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:title="@string/one"
        />
    <item
        android:title="@string/two"
        />
    <item
        android:title="@string/three"
        />
    <item
        android:title="@string/four"
        />
    <item
        android:title="@string/five"
        />
</menu>
```

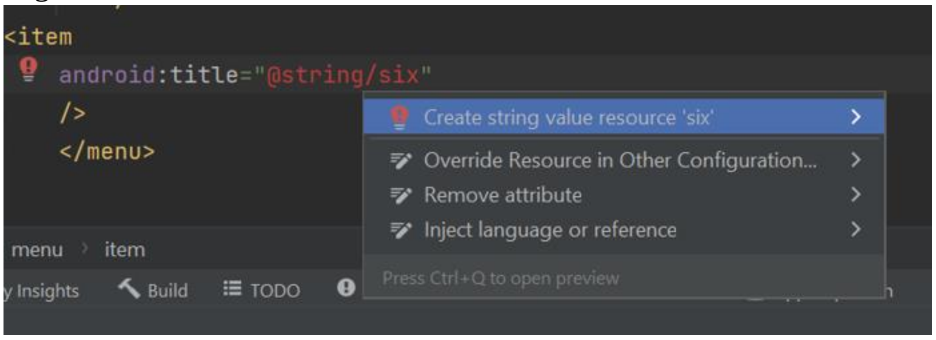
Note:

Output:



```
<item
    android:title="@string/six"
    />
    </menu>
```

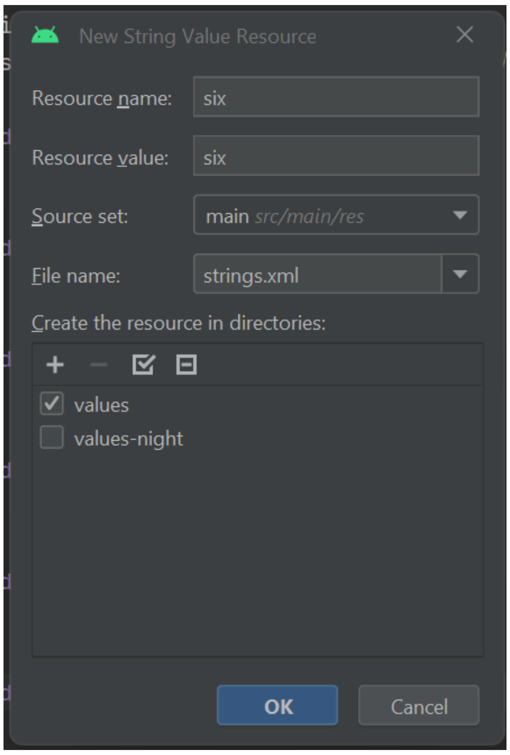
Right click



```
<item
    android:title="@string/six"
    />
    </menu>
```

Create string value resource 'six'
Override Resource in Other Configuration...
Remove attribute
Inject language or reference
Press Ctrl+Q to open preview
menu
item
y Insights
Build
TODO

Click on create string value resource


New String Value Resource

Resource name: six

Resource value: six

Source set: main src/main/res

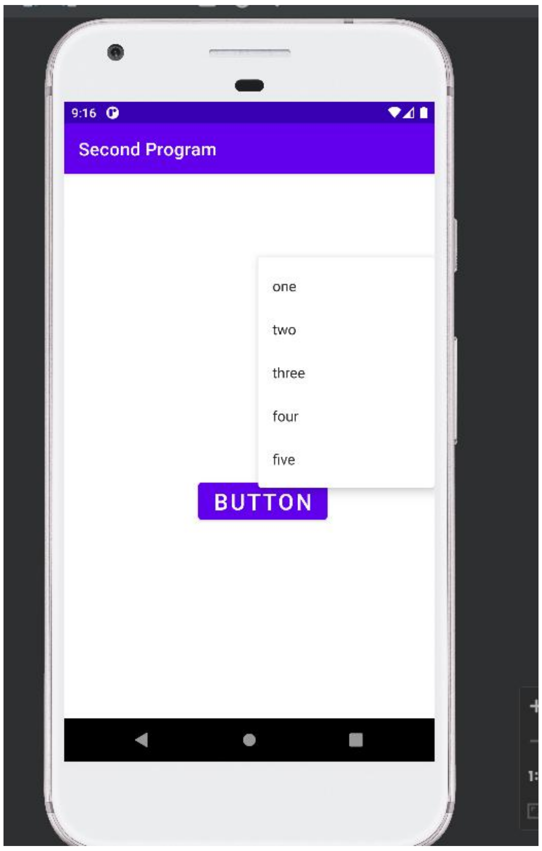
File name: strings.xml

Create the resource in directories:

- [✓] values
- [ ] values-night

Enter the Resource value and press **ok**

<u>Output</u>


9:16
Second Program
one
two
three
four
five
BUTTON