Lab 10: Android Styles

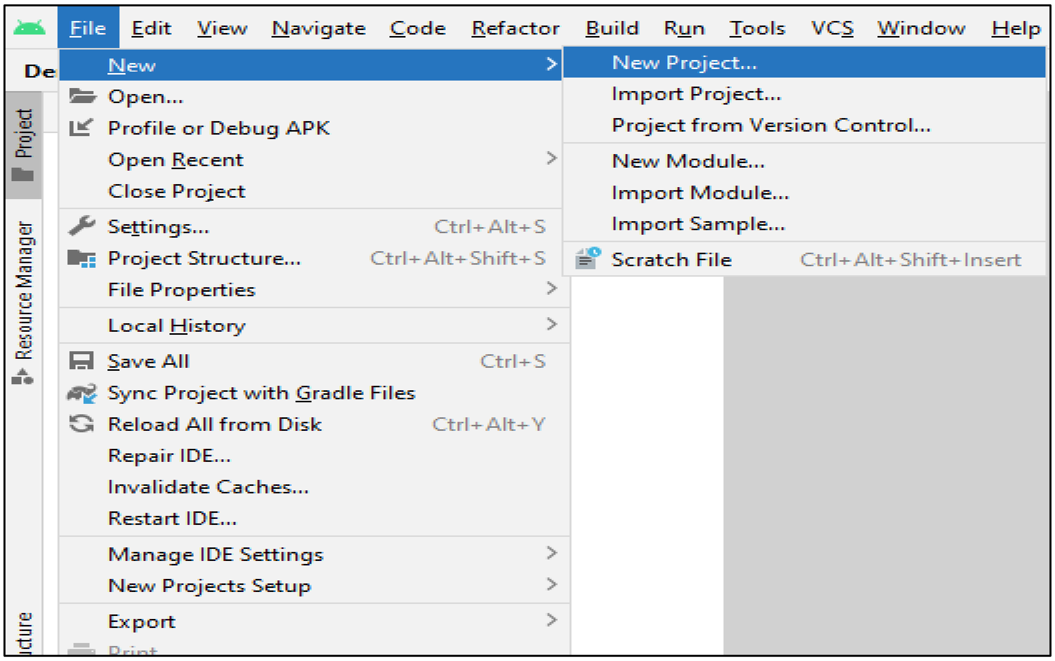
Introduction

A style resource establishes a UI's format and appearance. A view can have a style applied to it from within a layout file, and an activity or an entire application can also have a style applied (from within the manifest file).

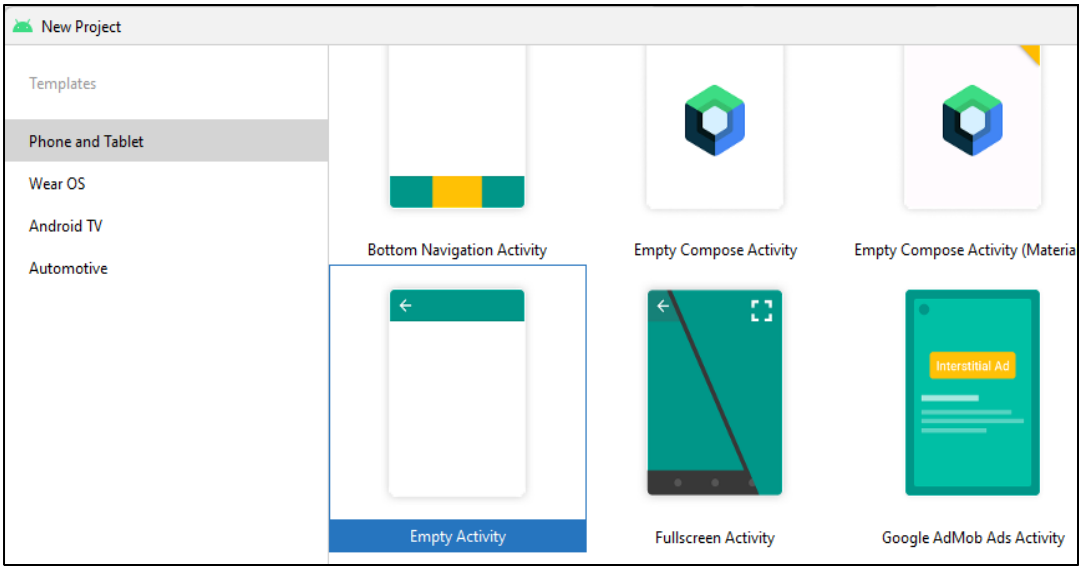
Let’s get Started

This exercise will take you through simple steps to show how to create your own Android application using Style.

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



Step 2: Select Empty Activity as shown below



New Project

Templates

Phone and Tablet

Wear OS

Android TV

Automotive

Bottom Navigation Activity

Empty Compose Activity

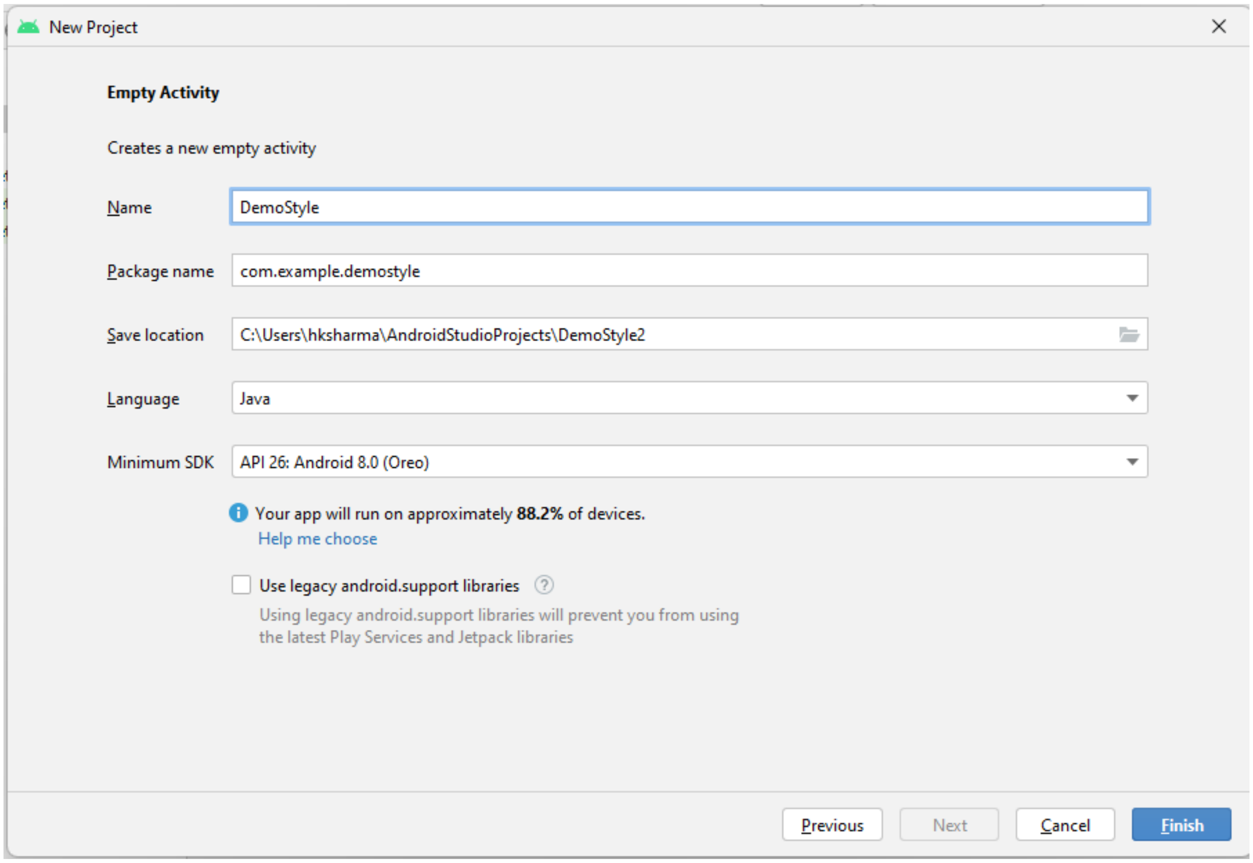
Empty Compose Activity (Materia

Empty Activity

Fullscreen Activity

Google AdMob Ads Activity

Step 3: Provide a Project Name as shown below



New Project

Empty Activity

Creates a new empty activity

Name: DemoStyle

Package name: com.example.demostyle

Save location: C:\Users\hksharma\AndroidStudioProjects\DemoStyle2

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

```
package com.example.demoStyle;
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: Create a new xml file style.xml in res/values folder and write the code given below

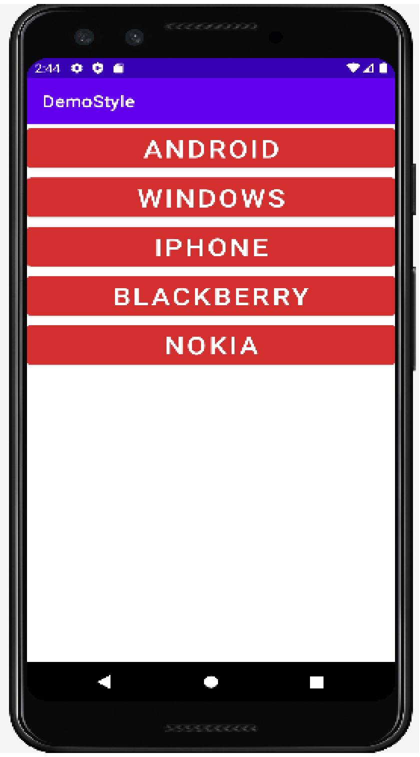
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="btnStyle">
        <item name="android:textSize">30dp</item>
        <item name="android:textColor">#fff</item>
        <item name="android:backgroundTint">#D32F2F</item>
        <item name="android:layout_width">fill_parent</item>
        <item name="android:layout_height">wrap_content</item>
    </style>
</resources>
```

Step 6: Update activity_main.xml as per the code given below

```
<?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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
    style="@style/btnStyle"
    android:text="Android"/>
    <Button
        style="@style/btnStyle"
        android:text="Windows"/>
    <Button
        style="@style/btnStyle"
        android:text="iPhone"/>
    <Button
        style="@style/btnStyle"
        android:text="Blackberry"/>
    <Button
```

```
        style="@style/btnStyle"
        android:text="Nokia"/>
</LinearLayout>
```

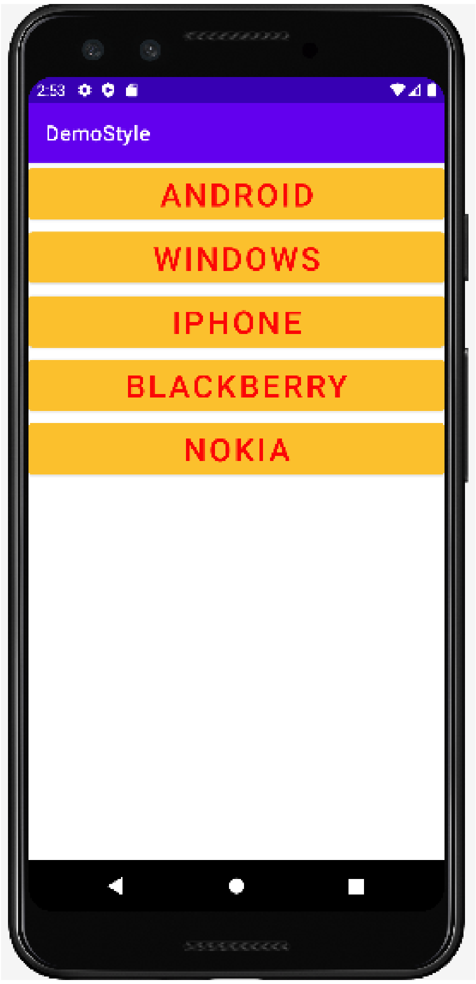
Step 7: Check Output on Android Emulator.



Step 8: Update style.xml as the code given below

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="btnStyle">
        <item name="android:textSize">30dp</item>
        <item name="android:textColor">#FB0606</item>
        <item name="android:backgroundTint">#FBC02D</item>
        <item name="android:layout_width">fill_parent</item>
        <item name="android:layout_height">wrap_content</item>
    </style>
</resources>
```

Step 9: Check Output on Android Emulator.



Voila!! We have successfully completed this lab.