Styles and Themes

▶ A style is a set of properties that have effects on how a View or Window looks like.

▶ These properties specify the font size, the font family, the background color, font color and so on.

▶ it separates the content structure from how it is displayed on the screen.

▶ A theme is a set of styles and can be applied to the whole app or to a single Activity. It helps to create an appealing user interface.

Styles and Themes

- Styles in Android allow you to define the look and feel, of Android components in XML resource files.

- This way you have to set common style attributes only once in one central place.

- Styles work by defining style names associated with a series of properties to apply to a view.

- Styles can also inherit from other style and compound styles can be created as well.

Styles and Themes

▶ To apply a consistent theme to all activities within our application.

▶ Instead of applying the style to a particular individual view, you can apply a collection of styles as a **Theme** to an Activity or application.

▶ When you do so, every View within the Activity or application will apply each property that it supports.

Styles and Themes

- You can create a theme the same way you create styles.

- The difference is how you apply it: instead of applying a style with the style attribute on a view, you apply a theme with the android:theme attribute on either the <application> tag or an <activity> tag in the AndroidManifest.xml file.

TextView Without a Style

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:background="#FF0000"
android:text="Hello style"
android:textColor="#FFFFFF"
android:textSize="14sp" >
</TextView>
```

Creating a Style

```xml
<resources
xmlns:android="http://schemas.android.com/apk/res/android" >
<style
name="MyTextStyle"
parent="@android:style/TextAppearance" >
<item name="android:textColor">#FF0000</item>
<item name="android:textSize">20sp</item>
</style>
</resources>
```

TextView without a Style vs. TextView with a style

```
<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Hello without style"

android:textColor="#FF0000"

android:textSize="14sp" >
```

```
<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Text with style"

style="@style/MyTextStyle"/>
```

Creating a Theme

```
<style

name="AllGreen"

parent="@android:style/Theme.Black" >

<item name="android:background">#00FF00</item>

</style>
```

and then we apply it to the application in the Manifest.xml:

```
<application

android:icon="@drawable/ic_launcher"

android:label="@string/app_name"

android:theme="@style/AllGreen" >

</application>
```

```xml
<resources xmlns:android="http://schemas.android.com/apk/res/android" >
    <style
    name="MyTextStyle"
    parent="@android:style/Widget.TextView" >
    <item name="android:textColor">#FF0000</item>
    <item name="android:textSize">14sp</item>
    </style>

    <style
    name="AllGreen"
    parent="@android:style/Theme.Black" >
    <item name="android:background">#00FF00</item>
    <item name="android:textViewStyle">@style/MyTextStyle</item>
    </style>
</resources>
```
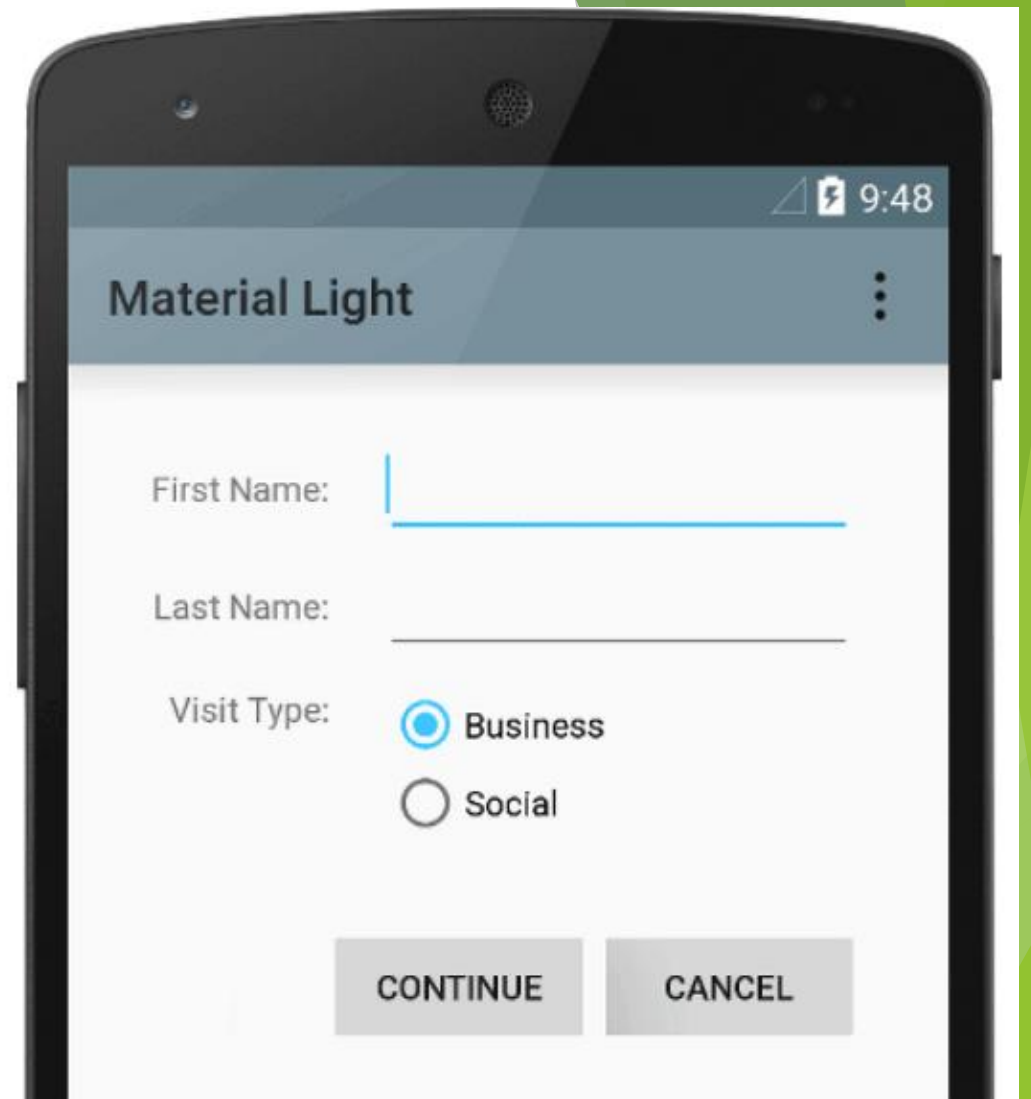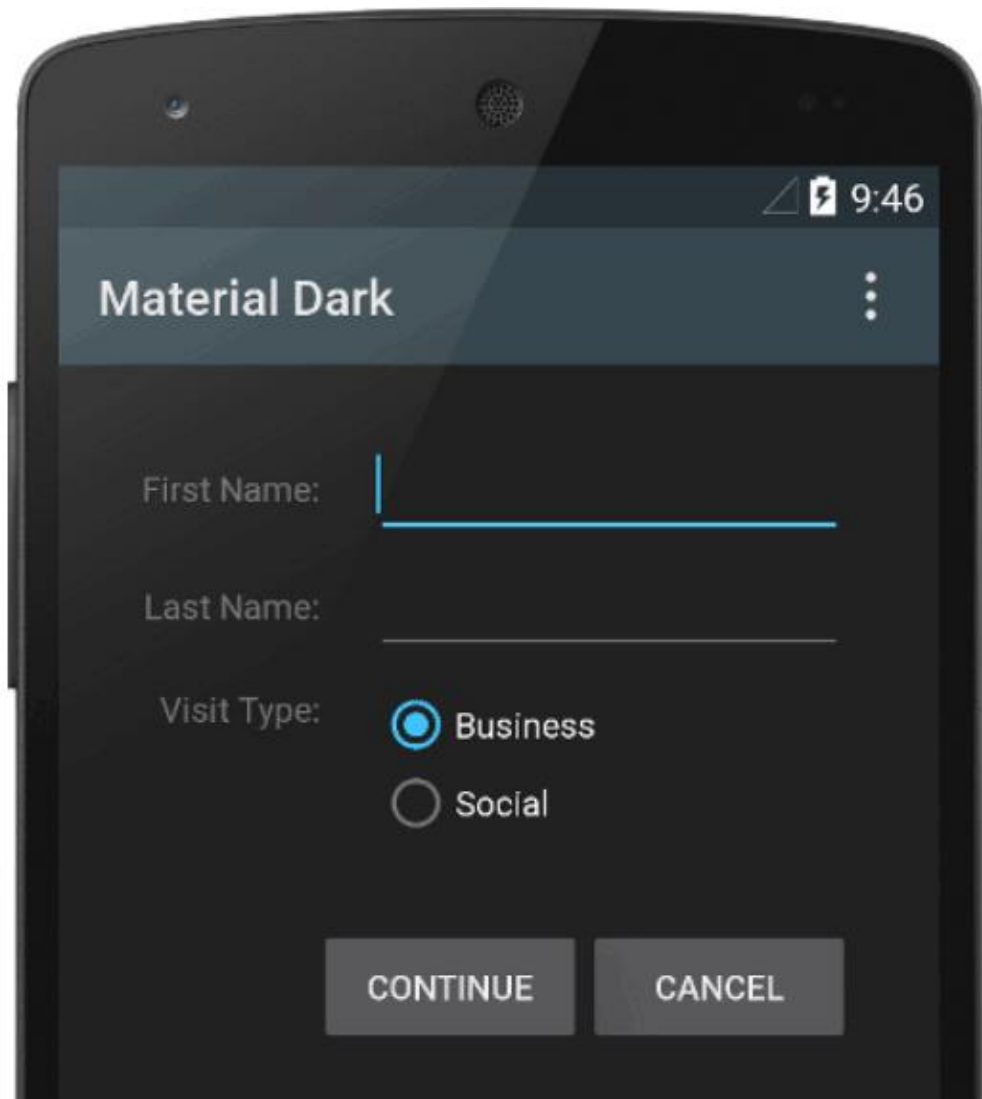
```xml
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>


<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!--  color for the app bar and other primary UI elements -->
    <color name="colorPrimary">#3F51B5</color>

    <!--  a darker variant of the primary color, used for
         the status bar (on Android 5.0+) and contextual app bars -->
    <color name="colorPrimaryDark">#303F9F</color>

    <!--  a secondary color for controls like checkboxes and text fields -->
    <color name="colorAccent">#FF4081</color>
</resources>
```

Theme.AppCompat (left) and Theme.AppCompat.Light (right)