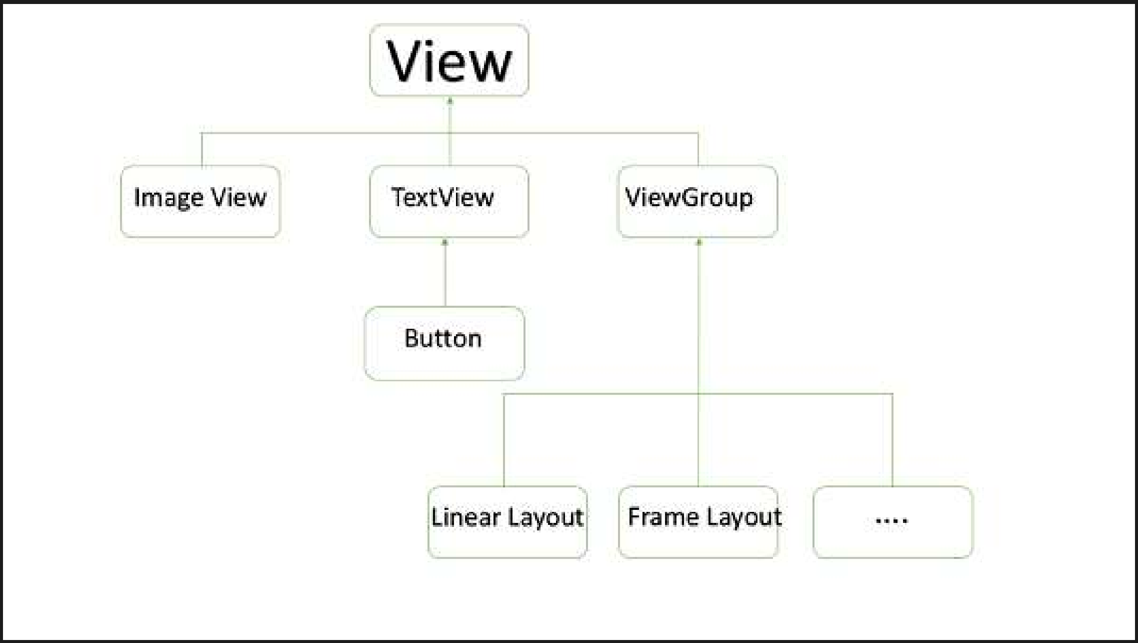
Android - Custom Components

Implementing own components in pre built-in components with extending subclass with own defined class

Android offers a great list of pre-built widgets like Button, TextView, EditText, ListView, CheckBox, RadioButton, Gallery, Spinner, AutoCompleteTextView etc. which you can use directly in your Android application development, but there may be a situation when you are not satisfied with existing functionality of any of the available widgets. Android provides you with means of creating your own custom components which you can customized to suit your needs.

If you only need to make small adjustments to an existing widget or layout, you can simply subclass the widget or layout and override its methods which will give you precise control over the appearance and function of a screen element.

This tutorial explains you how to create custom Views and use them in your application using simple and easy steps.



Example of Custom Components in Custom View hierarchy

Creating a Simple Custom Component

Step	Description

1	You will use Android studio IDE to create an Android application and name it as myapplication under a package com.example.tutorialspoint7.myapplication as explained in the Hello World Example chapter.
2	Create an XML res/values/attrs.xml file to define new attributes along with their data type.
3	Create src/mainactivity.java file and add the code to define your custom component
4	Modify res/layout/activity_main.xml file and add the code to create Colour compound view instance along with few default attributes and new attributes.
5	Run the application to launch Android emulator and verify the result of the changes done in the application.

Create the following attributes file called attrs.xml in your res/values folder.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <declare-styleable name="TimeView">
      <declare-styleable name="TimeView">
         <attr name="title" format="string" />
         <attr name="setColor" format="boolean"/>
      </declare-styleable>
   </declare-styleable>
</resources>
```

Change the layout file used by the activity to the following.

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

   <com.example.tutorialspoint7.myapplication.TimeView
      android:id="@+id/timeView"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textColor="#fff"
      android:textSize="40sp"
      custom:title="my time view"
      custom:setColor="true" />

   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/simple"
      android:layout_below="@id/timeView"
      android:layout_marginTop="10dp" />
</RelativeLayout>
```

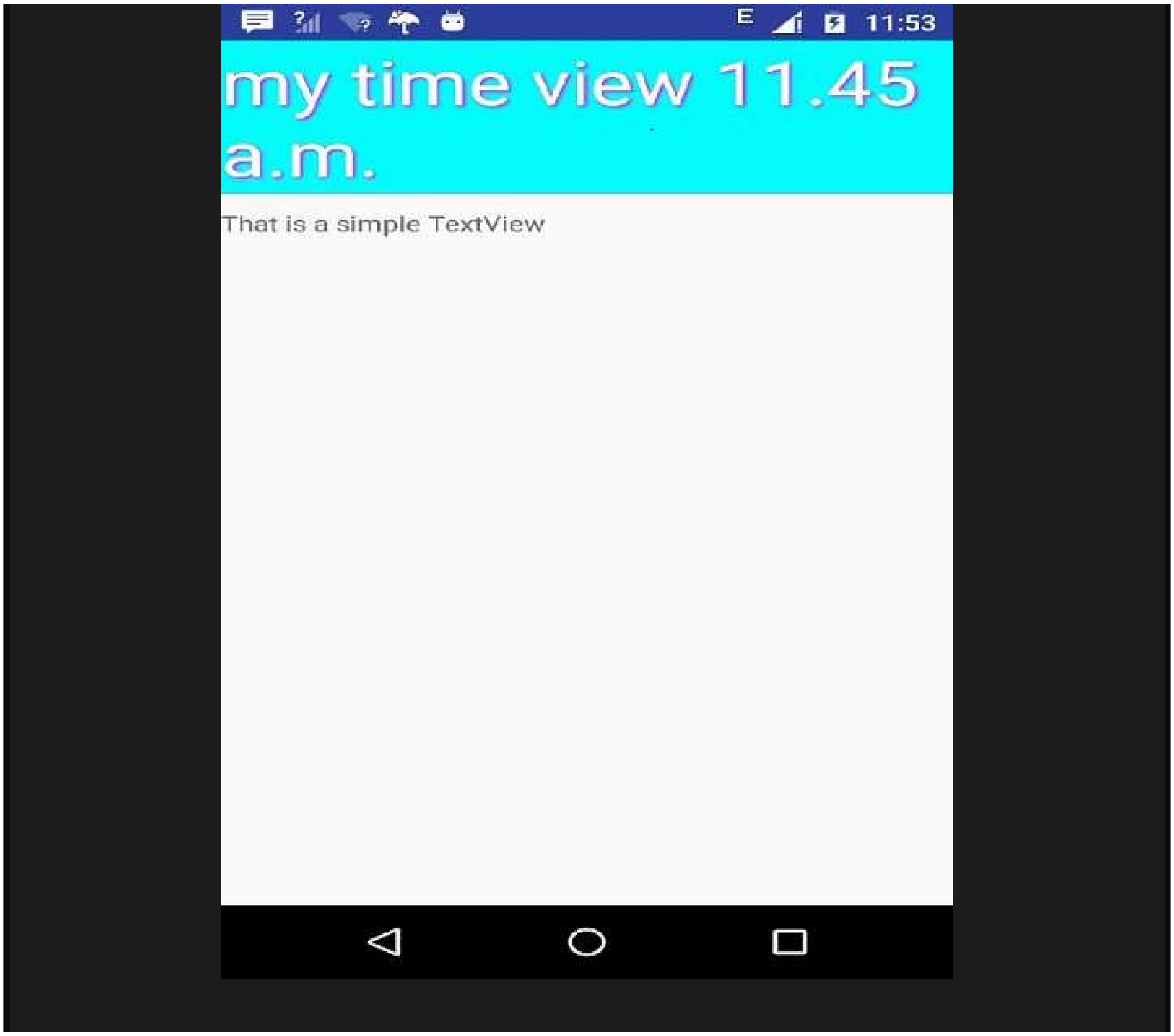
Create the following java file called timeview for your compound view.

```
package com.example.tutorialspoint7.myapplication;
/**
 * Created by TutorialsPoint7 on 9/14/2016.
 */
import java.text.SimpleDateFormat;
import java.util.Calendar;

import android.content.Context;
import android.content.res.TypedArray;

import android.graphics.Color;
import android.util.AttributeSet;
import android.widget.TextView;

public class TimeView extends TextView {
   private String titleText;
   private boolean color;

   public TimeView(Context context) {
      super(context);
      setTimeView();
   }

   public TimeView(Context context, AttributeSet attrs) {
      super(context, attrs);
      // retrieved values correspond to the positions of the attributes
         TypedArray typedArray = context.obtainStyledAttributes(attrs,
            R.styleable.TimeView);
      int count = typedArray.getIndexCount();
      try{

         for (int i = 0; i < count; ++i) {
            int attr = typedArray.getIndex(i);
            // the attr corresponds to the title attribute
            if(attr == R.styleable.TimeView_title) {
            
               // set the text from the layout
               titleText = typedArray.getString(attr);
               setTimeView();
            } else if(attr == R.styleable.TimeView_setColor) {
               // set the color of the attr "setColor"
               color = typedArray.getBoolean(attr, false);
               decorateText();
            }
         }
      }
        
      // the recycle() will be executed obligatorily
      finally {
         // for reuse
         typedArray.recycle();
      }
   }
   
   public TimeView(Context context, AttributeSet attrs, int defStyle) {
      super(context, attrs, defStyle);
      setTimeView();
   }
   
   private void setTimeView() {
      // has the format hour.minuits am/pm
      SimpleDateFormat dateFormat = new SimpleDateFormat("hh.mm aa");
      String time = dateFormat.format(Calendar.getInstance().getTime());
      
      if(this.titleText != null )
      setText(this.titleText+" "+time);
      else
         setText(time);
   }
   
   private void decorateText() {
      // when we set setColor attribute to true in the XML layout
      if(this.color == true){
         // set the characteristics and the color of the shadow
         setShadowLayer(4, 2, 2, Color.rgb(250, 00, 250));
         setBackgroundColor(Color.CYAN);
      } else {
         setBackgroundColor(Color.RED);
      }
   }
}
```

Change your Main activity java file to the following code and run your application.

```
package com.example.tutorialspoint7.myapplication;

import android.os.Bundle;
import android.widget.TextView;
import android.app.Activity;

public class MainActivity extends Activity {

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

      TextView simpleText = (TextView) findViewById(R.id.simple);
      simpleText.setText("That is a simple TextView");
   }
}
```

The running application should look like the following screen shot.


11:53
my time view 11.45 a.m.
That is a simple TextView