

Homework 04 – Attribute Setting Types and Resource Qualifiers

Due: Day 18 – March 14th, 2022 (beginning of class)

Objective:

The goal of this assignment is for you to demonstrate your understanding of how to programmatically set the 7 attribute setting types as well as apply resource qualifiers.

Instructions:

For this assignment, I have provided an Android Studio project on Moodle that you will need to download/unzip and use as your starting point. This project contains a single Activity (**activity_attribute_types.xml** and **AttributeTypesActivity.java**). The purpose of this homework is to setup a View Binding that will allow you to access the Views programmatically in Java code and alter their attributes that were selected to represent the 7 attribute types that we have discussed in class.

Important: You are not permitted to make any changes in the layout file (**activity_attribute_types.xml**) unless directly specified in the instructions below – otherwise, this will result in significant loss of points on this assignment.

Complete each of the tasks listed below in order in your Java code:

- **Task 0**
 - Perform the necessary steps to setup the View Binding feature for the project and add the necessary code to use the View Binding in the controller file
- **Task 1**
 - Complete the method named **task1()** in your Java controller file that sets the Rating Bar's star color (**progressTint**) to the following exact color named "MetroRed" found at [this website](#). Assume that we intend to use this color repeatedly throughout our app in the future and therefore, use the method that would be most appropriate for repeated future use. The Rating Bar method that you will need to call to set the progress's tint takes a **ColorStateList** as an parameter/argument – to convert an int that represents a color to a ColorStateList object, use the following static [ColorStateList.valueOf\(\)](#) method.
 - To test your method, call the task1() method in the onCreate() method of your Activity. When you click and drag your mouse to highlight stars, you should see the "MetroRed" color appear
- **Task 2**
 - Complete the method named **task2()** in your Java controller file that retrieves the text as a String from the EditText by calling **.getText().toString()** on it. Then, your code should set the 1st word in the String as the Toggle Button's text when it is in the OFF state. Next, your code should set the 2nd word as the Toggle Button's text when it is in the ON state. Tip: Explore the split() method in the String class for how to split a sentence into individual words (see this [tutorial](#) as an example).
 - **Important:** Do not hardcode "DePauw" and "Tiger" in your code as it will be worth 0% credit. Assume that I will enter two new words separated by a single space in the EditText when I test your code
 - To test your method, call the task2() method in the onCreate() method of your Activity. The Toggle Button will begin by displaying the word "OFF", however, once you click on the Toggle, its text should display the 2nd word in the ON state. When you click on the Toggle again, its text should display the 1st word now instead of "OFF"
- **Task 3**
 - Complete the method named **task3()** in your Java controller file that sets the gravity of the LinearLayout to center its Views horizontally and at the bottom.

- To test your method, call the `task3()` method in the `onCreate()` method of your Activity. The Toggle, Checkbox, and Switch should appear centered horizontally and at the bottom of the `LinearLayout`
- **Task 4**
 - Complete the method named `task4()` in your Java controller file that sets the appropriate attribute that causes all of the text in the `TextView` to be capitalized.
 - **Important:** You must accomplish this by using/setting the appropriate `TextView` attribute – do not extract the text and use `String` methods to perform this task as it will be worth 0% credit.
 - To test your method, call the `task4()` method and the `TextView`'s text should appear in all capital letters
- **Task 5**
 - Complete the method named `task5()` in your Java controller file that sets the Seek Bar's **thumb** ("slider icon") attribute to the **tiger.png** image file on Moodle. You must use the Android Drawable Importer plug-in to correctly add this image file into your project's resource folder.
 - To test your method, call the `task5()` method and verify that the slider (thumb) for the Seek Bar now displays the DePauw Tiger icon instead of a circle icon
- **Task 6**
 - Complete the method named `task6()` in your Java controller file that sets the **textAlignment** attribute of the `EditText` to **textEnd** (i.e., the text will be aligned to the right side of the `EditText` rather than the left side).
 - To test your method, call the `task6()` method and verify that the text (i.e., "DePauw Tiger") inside of the `EditText` is now aligned on the right side instead of the left side
- **Task 7**
 - Complete the method named `task7()` in your Java controller file that sets the Checkbox's **height** (not `layout_height`) attribute to **250dp** (not pixels)
 - To test your method, call the `task7()` method and verify that the height of the Checkbox is 250dp (i.e., larger than its original height in the layout file)

Complete **only** the following tasks in the layout file:

- **Task 8**
 - Use the appropriate method to display the following "education level" options in the Spinner:
 - Elementary
 - Middle
 - High
 - University
 - Next, use the appropriate method so that these options will automatically display in French when the user switches their language mode
 - To test your method, go to the AVD's Settings >> System >> Languages & input >> Languages activity. Then, select the + Add a language option and use the magnifying glass in the top-right corner to search for French and select France. When you make French (i.e., Français) the #1 language and switch back to your app, the Spinner's text should now display the education level options in French

Submission:

When you are finished, you must **zip** your Android Studio project. On Moodle, you should upload your zip file to the Homework 04 assignment box