CIS 3515 Lab

Worksheet 3

Instructions: You will create an application that will allow a user to see a list of colors in a drop down list (a Spinner view) where the background color for each list item will be the color that is listed in the associated text. When the user selects a color from the list, a second activity will be automatically launched that will receive the selected color from the first activity (via an intent extra), and use that value to set its layout's background color.

1. When creating your application, rename the main activity to PaletteActivity and your second activity to CanvasActivity.

2. PaletteActivity must contain a ListView that will use a **custom adapter** to present to the user a set of color options. The views generated by the custom adapter should be simple TextViews, where the text value is the name of the color, and the background of the text view should be the color specified by the name.

3. Your list of colors should be a predetermined (no fewer than 5 colors) array of strings. For simplicity choose colors from Android's Color class (e.g. Color.RED). This will make setting the background colors of views and layouts easier, since you only need to call the view's setBackgroudColor method with the color parameter. e.g.

```
if (chosenColor.equals("Red"){// where chosenColor is a string
    myLayout.setBackgroudColor(Color.RED);
} else if ...
```

Alternatively, if you choose basic colors, and store the values as simple strings, then the above code becomes simpler:

```
myLayout.setBackgroudColor(Color.parseColor(chosenColor))
// Color.parseColor(String) will only work for very simply colors
// see
```

https://developer.android.com/reference/android/graphics/Color.html#parseColor(java.lang.String)

4. When a user selects a color from the Spinner, the CanvasActivity should be launched and the layout's background should be set to the selected color (see illustration below)

5. Considerations:

    1. Remember that your custom adapter should extend BaseAdapter (although in this case you could, if you so choose, extend ArrayAdapter since the only thing we'll be changing is the view's background color – everything else would work the same

way an ArrayAdapter normally does).

2. You generally only need to implement the getView(), getCount(), and getItem() methods of your BaseAdapter subclass (if you extend some other classes, you may have to implement getDropdownView() for the spinner class to function).

3. Remember that to set a layout's background color from your activity's code at runtime (instead of XML at design time), the layout must have an ID so that you can reference it. Add an ID field to the layout at design time if one wasn't added for you automatically. Pay attention to the syntax as it's easy to get it wrong.

4. Note that when the Spinner first loads, the first element that is shown automatically triggers the "Selected" listener. Investigate ways to avoid this unwanted "feature".

6. **You have one week to complete this assignment. Upload your project or a GitHub link to Canvas, or make arrangements to show your work to the TA <u>BEFORE</u> the beginning of next week's lab**.

   ILLUSTRATION: