CIS 3515 Assignment 3

Instructions: Create an application that uses a custom adapter. Your application will allow a user to select a color from a **Spinner** and once selected, set the background of the activity's layout to the selected color.

1. Create a **custom adapter called ColorAdapter** that extends **BaseAdapter** that will 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. The background of the text view should behave as follows:

i. When the spinner is in *drop-down* mode (i.e. the user clicked the drop-down button and the various selection options are being displayed), each textview's background color should be the same as its displayed text value.

ii. When an item is selected, the view's background color should be set to white.

2. When the user selects a color from the Spinner, the activity's **layout** background should be set to the selected color. One way to change the background color of the layout, which you will recall is also a view object, is to assign it an ID as you would any regular view object. Then from within your code you can access the layout by passing its ID to *findViewById(R.**id**.layout_id),* and from there change it's background color.

3. Your list of colors should be a predetermined (no fewer than 10 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.:

```
myLayout.setBackgroudColor(Color.parseColor(chosenColor))
// Color.parseColor(String) will only work for a predefined set of color names
// see
https://developer.android.com/reference/android/graphics/Color.html#parseColor(java.lang.String)
```

4. When your application first starts, no color should be selected (the layout's background should not be changed)

5. **Push your project to GitHub**.

Rubric

Application uses custom adapter and implements all relevant functions	20%
Custom adapter does view recycling	20%
Views are different for each spinner state (Colored when dropped down. White when selected)	30%
Activity layout color is changed when color selected	20%
Activity color is not changed when application first starts	10%