CIS 3515 Assignment 4

Instructions: Create an application that allows a user to pick an image from a **RecyclerView** and once selected, display the selected image inside the main activity along with a description.

1. Add a collection of images to your project as **drawable** resources matching a theme of your choice (e.g. Dogs, Cats, Mountains, Comic Book characters, etc.). You should have no fewer than 10 images. You should also have a textual name/description for each of the images in your application.

   *Try to use images with a relatively small file size (about 800kb or less per image). If your images are too large, your application may crash with an **OutOfMemory** error.*

2. Add a RecyclerView, an ImageView and a TextView to your activity's layout, ensuring that all views are nicely arranged.

3. Your RecyclerView should use a **GridLayoutManager** so that its views are displayed in a grid. You can decide on the number of columns you would like to have, as long as it's more than 1.

4. Create a **custom adapter** called **ImageAdapter** that inherits from **RecyclerView.Adapter.** This adapter will generate views containing the drawables you added to your project earlier. The views generated should be relatively small; approximately 200x200 pixels, but you can adjust that size based on how many columns your RecyclerView has.

5. When the user clicks an item in the RecyclerView, the activity should display a larger version of the selected image in its ImageView, along with the item's name/description in its TextView.

6. **Be sure to make commits as you progress and to push your project to GitHub after each commit – This is necessary to create an audit trail of your work. You can and *will* lose points if this is not followed.**

   HINTS and CONSIDERATIONS

   Add images to your project as resources using the resource manager:
   https://developer.android.com/studio/write/resource-manager#import

   When an image (example: **test_image.png**) is added to your project as a resource, you can reference it in your code as follows: R.drawable.*test_image*

   All resource references (example: R.drawable.*test_image*) are ***Integers***

   To display that image in an ImageView: myImageView.setImageResource(R.drawable.*test_image*)

   To keep track of your images and associated descriptions, you should consider creating a data class for each element (resourceId: **Int**, description: **String**), and then build an array of those objects

Rubric

Images added to project as drawable resources	10%
RecyclerView uses GridLayoutManager for layout	10%
Application creates custom instance of RecyclerView.Adapter and implements all methods	40%
Event listener is implemented for views in adapter	30%
ImageView **and** TextView in Activity updates when RecyclerView item is clicked	10%