COMP 10073 - Android Application Development

Lab Assignment 3

Overview:
Create an application with Fragments which preserves state when restarted.

Assignment Requirements:

- Create a new Android Application using the EMPTY Template. Select API 28 (Android 9.0) as the Minimum SDK. You **must** use a domain naming pattern of **ca.mohawk.yourname** when you create your project (last name is sufficient). You must also include the following statement of authorship in a comment at the top of your MainActivity java code:

  > **I, *John Doe*, *000123456* certify that this material is my original work. No other person's work has been used without due acknowledgement.**
  > *(Replace with your own name and student number)*

- The Application should be named "Lab3", however the displayed name ("**app_name**") should be changed to the following:

  Lab 3 - Your Name 000123456

1. Your MainActivity will use a SharePreferences object to store a number which will be the last value of your counter.

2. When your app starts, read the SharedPreferences value - if there is nothing stored, default to zero. Update the value just before the application stops.

3. Create two Empty Fragment Java classes - **`TopHalf`** and **`BottomHalf`**

4. For the Main Activity, create a Layout with the following elements:
   a. Two FrameLayouts to hold the Fragments.
   b. The top fragment should use 25% of the UI, the bottom 75%.
   c. Assign your TopHalf and BottomHalf Fragment classes to the frames.

5. Use the fragment manager to load the fragments into the FrameLayouts

6. In the Layout for the TopHalf fragment, add two buttons : [-] and [+]
   Colour the background of this fragment with colour value #FFE0E0
   The buttons should be vertically (up/down) centered in the layout.

7. In the Layout for the BottomHalf fragment, add a TextView to show the current count. Colour the background of this fragment with colour value #E0E0FF
   The TextView should be vertically and horizontally centered in the layout.

8. When a Button is clicked, add or subtract from the current count, and display in the TextView. Do not save the value to shared preferences each time a button is clicked - this creates terrible performance issues in complex applications.

9. You should define a new "Drawable Resource File" XML. See below for sample XML code. Apply this custom drawable shape to the buttons in your Layout, ie:

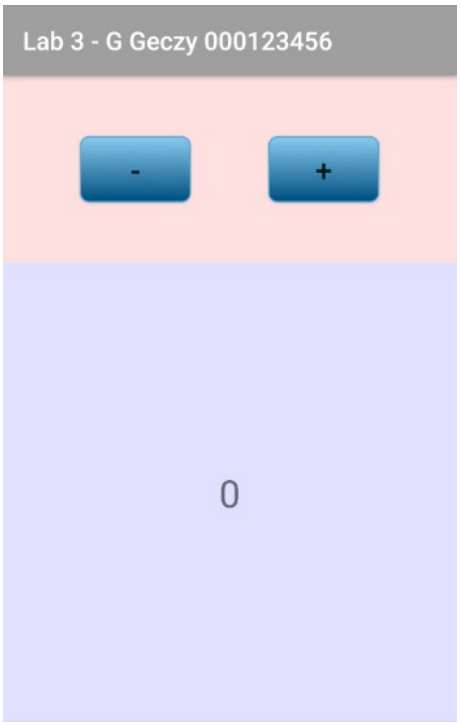
    `android:background="@drawable/custombutton"`

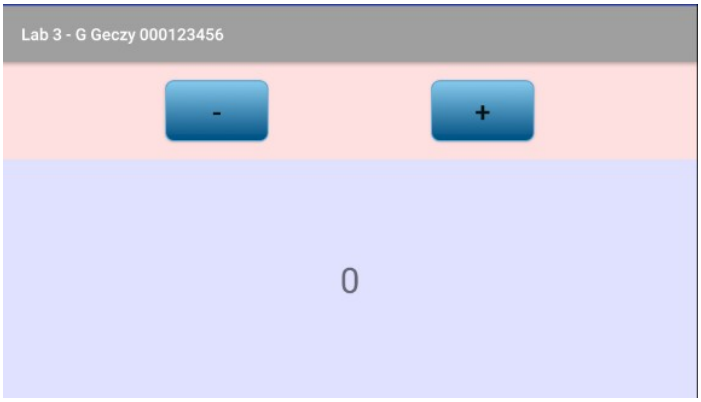
10. When your App is closed and re-started, it should remember the previous number value (load the value from SharedPreferences).

Example Drawable Resource File *(modify / change for yours, and add a "state_pressed" version)**:***

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
   <item>
      <shape>
         <gradient
            android:startColor="#89cbee"
            android:endColor="#004F81"
            android:angle="270" />
         <stroke
            android:width="1dp"
            android:color="#4aa5d4" />
         <corners
            android:radius="7dp" />
         <padding
            android:left="10dp"
            android:top="10dp"
            android:right="10dp"
            android:bottom="10dp" />
      </shape>
   </item>
</selector>
```

Example displays :





Resources:

- Android Developer Guide - Fragments
  - https://developer.android.com/guide/components/fragments.html
- Storage Options / Using Shared Preferences
  - https://developer.android.com/guide/topics/data/data-storage.html
- Shape Drawable
  - https://developer.android.com/guide/topics/resources/drawable-resource.html

Demonstration Video

Record a 25-45 second demonstration video. In your video:

- Discuss how you used the fragment manager to load fragments programmatically.
  - You *must* use the fragment manager to get full credit for this lab.
- Demonstrate the use of your app. Confirm that when started it loads old values.
- You must use a microphone to record your answers along with desktop capture.
- Upload your video to a file sharing service like YouTube.
- Edit your video if it is too long. Keep it between 25secs and 45secs. See:
  a. How to Edit Videos with YouTube. https://youtu.be/84uLZ2pPebI

Submission

- At the top of your MainActivity.java file Include a link to your demonstration video in a comment below your statement of authorship.
- Use Android Studio to Export your project, including the .java code, related .xml files, and all other project elements to a ZIP archive.
  - Your ZIP archive should be a few 100K at most.
- Upload your .zip file to myCanvas by the deadline.