Project 5: ComplexCounter

Release date: **Friday, April 6th, 2018**

Due date: **Sunday, April 22nd, 2018 by 11:59pm**

This is an individual project, to be completed on your own. It is considered dishonest either to read someone else's solution or to provide a classmate with a copy of your work. Do not make the mistake of thinking that superficial changes in a program (such as altering comments, changing variable names, or interchanging statements) can be used to avoid detection. If you cannot do the work yourself, it is extremely unlikely that you can succeed in disguising someone else's work. We are adamant that cheating in any form is not tolerated. Even the most trivial assignment is better not done than if you cheat to complete it.

Prerequisites

1. Understanding of the Android Studio interface
2. Ability to exchange data between activities
3. Ability to reference and modify layout elements

Learning Objectives

1. Utilize Android Studio to create mobile applications
2. Create Activities for separating functionalities
3. Write code that manipulates multiple layout elements
4. Create Android app layouts

Problem

Mobile applications make up one of the greatest consumer markets in our digitalized world. Most Android apps in particular are developed using Java along with an extensive framework to help developers get their ideas out on the market. As you will probably notice, not a lot of these ideas are original (*cough* SurvivalCraft *cough*), but you have to get your start somewhere.

In this project, you will be creating a basic app that you too can flood the market with. Seeing that often there are many "things" that are countable and you only two or fewer hands to count on, you decide to make an app to help people count things. Your app will allow people to create lists of things to count. A mock-up of the app screens and behaviors is provided in the zip file containing this document.

Activities

Your app will consist of three activities, named as such:

- **MainActivity**: The main activity for your program will be a list of the counters the user has created. These counters should persist throughout the duration of the program's execution unless they are deleted. The counter listing will also provide a button for incrementing the value of each of your counters at once through use of the "Auto" button.
- **AddNew**: The user should be capable of creating new named counters for whatever they need counted. Counters created here will be added to the list of counters in the Counter activity.
- **Counter**: The user should be capable of changing the value of their counters. Selecting a counter will present the user with this activity, giving them the option to change the value of the counter by either incrementing it by one or assigning a manually inputted value.

Considerations

When developing your app, please keep these factors in mind:

- **Your app will be graded on a Moto G provided by the Computer Science department**. They will be run on the 6.0 version of Android (Marshmallow).
- We will be grading your apps in lab. **If you will not be able to attend your lab session during the week of April 23rd, please contact your GTA about arranging a grading session**.
- **Hint**: For transferring information between activities, consider how you can represent your data as a data type that can be passed using putExtra. There is no one specific way to do so.
- **Hint**: Do a little research on **ListViews** to figure out how you can populate the list in MainActivity. There are some tutorials out there on the topic.

Submission

For this assignment, you will be required to submit:

- Java Files: You should submit the Java file corresponding to each of the activities you create and other .java files you create that are required by your activities (if any).
- XML Files: You should submit the files in res/layout. Ensure that the files for each activity are included.
- APK: To help speed up the grading process, you will need to submit a debug APK which will let us run your program quickly. To get the APK, in Android Studio go to *Build -> Build APK(s)*. Your file, app-debug.apk, should be in the "app\build\outputs\apk\debug directory" of your project. Submit this to Vocareum.

You will also be required to bring your Moto G to your lab session should it be needed for grading. Because we must prepare for grading in lab, **you will be unable to use a slip day for this assignment**.

Grading Rubric

The following are the required features for the activities of your project. They will not necessarily be tested in the order as presented, so ensure you test with variety.

Description	**Value**
MainActivity: "ADD NEW" and "AUTO" buttons present	5
MainActivity: "ADD NEW" button changes current activity to AddNew activity	10
MainActivity: "AUTO" button increments all counters by 1	10
MainActivity: ListView present, initially displays a single counter	10
MainActivity: Selecting a counter name in the ListView changes current activity to Counter activity.	10
AddNew: Text field present, can be edited by user	5
AddNew: "SET" button present, changes current activity to MainActivity	5
AddNew: After pressing "SET" button, new counter with specified name is added to MainActivity	10
AddNew: After pressing "SET" button, previous counters are still present in MainActivity	10
Counter: Text field present, can be edited by user	5
Counter: "COUNT" button present, increments text field value by 1	5
Counter: "DONE" button present, changes current activity to MainActivity.	5
Counter: After pressing the "DONE" button, the counter selected from MainActivity has a new value based on the final value in the Counter's text field. MainActivity is unchanged otherwise.	10