Android Coding Challenge:
Basic UI and Concepts

Instructions

You should be able to finish this challenge in 100 minutes or less. It is broken down in the parts described in the following section. The app should work in both portrait, and landscape orientation. **Make sure to commit and push code to a public GitHub repository**.

Good luck!

App Summary

You will create a very simple list-based app. It will have exactly one *Activity*, and should not use any custom Views. Everything will be done using the standard Android framework, with the addition of the Android design library and *RecyclerView* library.

The app will function as follows…

Part 1: App Setup (5 minutes)

1. Create a new Android Studio project, and name it anything you like. Make sure that the app supports API level 19 and above.
2. Add an Activity to your project called *MainActivity*
3. Add your dependencies for *RecyclerView* and the Android Design library

Part 2: Create a list (30 minutes)

1. You will display a list of names. The design can be as simple as you like, as long as the first and last name of the person is split like so: “Simon, Ryan”. Last name, First name.

2. Use dummy data to populate the list with at least 5 unique names. Setup your dummy data as if you were populating it from a JSON-based API.

Part 3: Adding new names to the list (45 minutes)

1. Add a *FloatingActionButton* that is anchored to the bottom right corner of the *RecyclerView*. Add a background color to it, and the following image.
2. On click of the *FloatingActionButton*, a *DialogFragment* should appear, and allow the user to type in a name. When the user hits “ADD” on the dialog, the dialog should be dismissed, and the typed in name should be added to the end of the existing list of names.
3. When adding a user’s name to the list, make sure that even if the name is added in all lower case, that it is still displayed capitalized. i.e. input = “simon, ryan”, displayed = “Simon, Ryan”.

Part 4: Conceptual Questions (20 minutes)

Please explain the following concepts. Make sure to **include concrete examples** to back up your explanations. If you don’t know the answer, simply write “I don’t know”. **All answers should be written in Markdown as part of the README file located in your code challenge repository.**

1. What is the major difference between an *abstract* class and an *interface*?
2. Why is Java 7’s class inheritance flawed?
3. What are the major differences between *Activities* and *Fragments*?
4. When using *Fragments*, how do you communicate back to their hosting *Activity*?
5. Can you make an entire app without ever using *Fragments*? Why or why not? Are there any special cases when you absolutely have to use or should use *Fragments?*
6. What makes an *AsyncTask* such an annoyance to Android developers? Detail some of the issues with *AsyncTask*, and how to potentially solve them.