Stanford University, CS 193A
Homework 1: First Android App

The purpose of this assignment is to get a first taste of Android application programming and using the Android Studio editor, as well as using widgets/views and events to produce an interactive graphical application.

Before working on this or any assignment, you will need to set up the **Android Studio** IDE on your computer. We suggest that you do this as early as possible, because some students get stuck and need time to get the IDE set up.

Assignment Description:

Flexibility: Compared to a course like CS 106A/B/X, the assignment specs for this class will be much more relaxed and vague. The idea is to encourage you to use your creativity to build something that sounds interesting to you, rather than forcing you to produce exactly a certain app or outcome. Some assignments may have some constraints or fixed aspects, but even these can be somewhat flexible if you have a neat idea that falls outside the guidelines.

Constraints: For this first assignment, we'll leave things very open-ended. You can make **any Android app you want** and submit it to receive credit. We'd prefer to see an app that has the following qualities:

- Your app should be set up as an **Android Studio** project, so it can easily be opened/run/graded by others.
- Your app should use at least **4 widgets**/views on the screen. *(Example: Three Buttons and a TextView, ...)*
- Your app should use at least **2 different kinds of widgets**. *(Example: Button, TextView, EditText, ...)*
- Your app should respond to at least two different **events**. *(Example: clicks on two different buttons.)*
- Your app's design should change at least some aspect of the physical appearance and **styling** of some widgets. For example, make a text view have a larger or bold font, or make a button appear in a blue color.
- We haven't talked very much about layout yet, so your app can have essentially any **layout** you want, so long as the various widgets are visible and can be interacted with.
- Along with your app, please turn in a file named **`README.txt`** that contains your name and email address along with the name of your app and a very brief description of it, along with any special instructions that the user might need to know in order to use it properly (if there are any). For example:

```
Joe Student <jstudent@stanford.edu>
NumberGame 2.05 - This app shows two numbers on the screen and asks
the user to pick the larger number.  Perfect for Berkeley students!
Note: Runs best on big Android tablets because of 1000dp font choice.
```

Suggestions:

Outside of the constraints on the previous page, you can do anything you want! Again, we're trying to let you be creative here. Your app does not need to be totally unique; if you have an existing idea that you want to replicate, or if you and a friend want to make the same kind of app, please go ahead.

If you are not feeling very inspired and want some suggestions of what to make, here are a few ideas. Some of the ideas below are trickier than others or use material we haven't covered yet, so you'd have to do your own digging.

- **Random number guessing game:** The computer thinks of a number from 1 to 1000. The user makes guesses, and after each incorrect guess, the app hints to the user whether the right answer is higher or lower than their guess. The game ends when the player guesses the right number.
- **Tip calculator:** User types in how much money they spent on their meal at a restaurant and chooses a percent to tip, and the app outputs how much money to leave as the tip.
- **Grade computer:** Type in your homework and exam scores and it estimates your grade in a class.
- **Hangman:** Computer thinks of a word and the user has to guess letters to try to reveal it.
- **Memory game:** Several buttons are shown on screen, "face down", and the user needs to try to find pairs that match up when flipped over.
- **Rock-Paper-Scissors:** The human and computer players each pick an option of Rock, Paper, or Scissors. The computer's choice is made randomly. Paper beats Rock; Scissors beat Paper; and Rock beats Scissors. Keep track of the human player's score against the computer over time.

We recommend you do something simple for this first program since you will have a lot of new syntax and features to get used to. These assignments, as well as this class in general, are meant to be **low-stress** and fun. If you want help, please feel free to show your code to others or ask for help in our online **Piazza forum**. Feel free to make an app as simple or as complex as you like, relative to your familiarity level and time constraints. If you work on your solution for roughly 2 hours and are still not done, you can turn it in and we will award you credit. You can do it!

Turnin and Grading:

Instructions for turning in this program can be found on the class web site. After programs are turned in, you will be asked to peer-evaluate another student's submission. Details will be provided later.

Your submission will be graded quickly by simply running it and evaluating its functionality. It does not need to be perfect or bug-free to receive credit. Your code will not be graded on style, but we still encourage you to follow good overall coding style for your own sake. If you want to see some good examples of proper Java coding style, consult the **Style Guide** linked from the Homework web page.