

## CIS 3515 Assignment 2

### Instructions:

Create an Android application to collect a user's **name**, **email**, a **password**, and a **password confirmation** on its main activity. It should also contain a **Save** button.

When the user clicks the Save button, it will display a message containing the user's name, welcoming them to the app. e.g. **"Welcome, Jayne Dough, to the SignUpForm App"**

As always, please ensure that your application has a proper Git commit trail that shows your progress throughout the development process. It is not sufficient to only have a single commit with all your code at the end of the project, and you may be penalized if that is the case.

1. Create a new *Application* called **SignUpForm** - use **edu.temple** for the project package
2. Rename your *Activity* using the **Refactor** option. Name your activity **FormActivity**  
(HINT: Refactoring a class name is a big change to your program, so it would be a good idea to create a Git commit before and after)
3. Use appropriate views to create a regular *sign up form* (similar to what you might see in an app or website when you create an account) that will collect the information detailed above.
  1. Use an appropriate text field for each element (e.g. to accept an email address, simply use an EditText view with the **inputType** attribute set to *textEmailAddress*).
  2. For each input field, you should set the **"hint"** attribute to indicate to the user what value should be entered there
4. When the user enters the information and clicks the save button, your application should do the following:
  1. Make sure that every field has data entered
    1. If any field is empty display an Error Message for each field to tell the user they must enter that information
  2. Compare the **password** and **password confirmation** to make sure they match
    1. If they don't match, display an Error Message telling the user to fix this error
  3. If all fields are complete and the passwords match, display the user's name inside a welcome message similar to the example given in the Instructions using a **Toast**.
  4. **All error messages must be displayed using the "error" attribute of the associated EditText**  
([https://developer.android.com/reference/kotlin/android/widget/TextView#setError\(kotlin.CharSequence\)](https://developer.android.com/reference/kotlin/android/widget/TextView#setError(kotlin.CharSequence)))
5. **Push your Project to GitHub and post the repository URL to Canvas. Ensure that the TA has access to your project.**
6. Add a **README.md** (<https://help.github.com/en/articles/about-readmes>) file to your GitHub project that details the function of your application. Include a screenshot in your readme. You can retrieve a screenshot using the Android emulator.

## Rubric

Layout is symmetric and clean	10%
Each input field uses the proper inputType attribute	20%
All error messages are displayed per requirements	40%
Welcome message displayed when successful	20%
README.md is present with screenshot on GitHub	10%