CPSC x150 Mobile Device Software Development **FA19**
Homework 1 – 30 points

You are highly recommended to start early on this project since you need to set up and become familiar with a new IDE. Ask questions on Piazza! Don't wait until the last minute!

Assignment Objectives:

- Install and create a basic project using Android Studio.
- Use the Android Studio device emulator
- Create a basic project that involves an activity, `onClick` events, layouts, and drawable resources.
    - Don't worry if you notice that later zyBooks chapters go into much more detail of the above topics---what you need you can get from zyBooks 1 and 2 and the associated code from those chapters.

Delivery Instructions:

- This assignment is due by 11PM on **Friday, September 6th, 2019** and will not be accepted late.
- Upload a .zip file containing the following to appropriate assignment on Canvas:
    - Your Android project directory containing all source code.
    - A readme.txt containing your name, CID number, and directions on how to load and run your program.

Approximate Grading Distribution:

- Test Cases (20 points): Checking to see if your program functions efficiently and correctly on an Android Emulator simulating a Google Pixel 2 device running Oreo 8.0. Code should compile cleanly (e.g., without errors or warnings). All code should be written in Java (not Kotlin).
- Proper Documentation (5 points): Code should have consistent and proper indentation. Functions should have appropriate comments (including pre/post comments). Any code inspired by zyBooks or the official Android tutorials should have clear citations within your code comments (both in the header and where the code is used). Your name, CID, and email address should appear in a header in each file.
- User Interface / User Experience (5 points): App has a nicely-defined user interface as per guidelines discussed in class and/or readings.

Collaboration Policy:

- For this assignment, you should work on developing your code yourself.
- You may consult the following resources:
    - zyBooks Chapters 1 and 2
    - https://developer.android.com/guide (including Code Labs)
    - https://docs.oracle.com/javase/8/docs/api
- You may use Piazza for any of the following:
    - Asking conceptual questions about how your code functions.
    - Referencing code from zyBooks, Android's Developer's Guide, or the Java API.

Part 1: Hello, Android Studio!

In computer science education, the typical first program a student writes is, "Hello world!" (see: https://en.m.wikipedia.org/wiki/%22Hello,_World!%22_program). The purpose of this assignment is for you to create your own Hello World program, but with a little more flair and polish. However, we first need for you to set up Android Studio and create a working emulator.

1. Install Android Studio on your personal laptop from: https://developer.android.com/studio
    a. Note, since Android's source code is in Java, you will need to also install the Java Development Kit (JDK) (if it's not already installed on your laptop.) Some developers recommend installing the JDK prior to installing Android Studio.
        i. http://www.oracle.com/technetwork/java/javase/downloads/

ii. We recommend installing Java 8 for this class. Android Studio supports all Java 7 language features and a subset of Java 8 (see: https://developer.android.com/studio/write/java8-support)

2. Create a "Hello World" project to make sure your Android Studio is correctly set up and functioning as expected.
    a. You will be using all the default settings for a Hello World project.
    b. When you get to "Add an Activity", choose the default "Empty Activity" and go ahead and create the project. Make sure you select "Java" as your language.
    c. The default view will show several toolbars, palettes, and a generic Android devoice.
    d. Press the Android Virtual Device button at the top of Android studio and create a Virtual Device with the category set to "phone", "Device Name" to "Pixel 2", and choose "Oreo 8.0" for the System Image.
        i. Note that Android Studio will give you a warning about how few devices will actually run. This ties into lecture with respect to the fragmentation of the Android ecosystem.
        ii. Note, you may need to click on download next to "Oreo" in order to install the necessary support files to emulate a Google Pixel 2 phone.

    Note, some Windows computers may show an error stating that x86 emulation requires hardware acceleration and that a "HAX kernel module" is not properly installed. To fix this, you can either:
    1. Follow Google's directions on enabling virtual machine acceleration (http://developer.android.com/tools/devices/emulator.html#vm-windows)
    2. Switch to ARM CPU mode rather than x86, however, your emulator will run very slow.

Part 2: Individual LinkedIn

All students will create an "Individual LinkedIn" Android application with the following specifications:

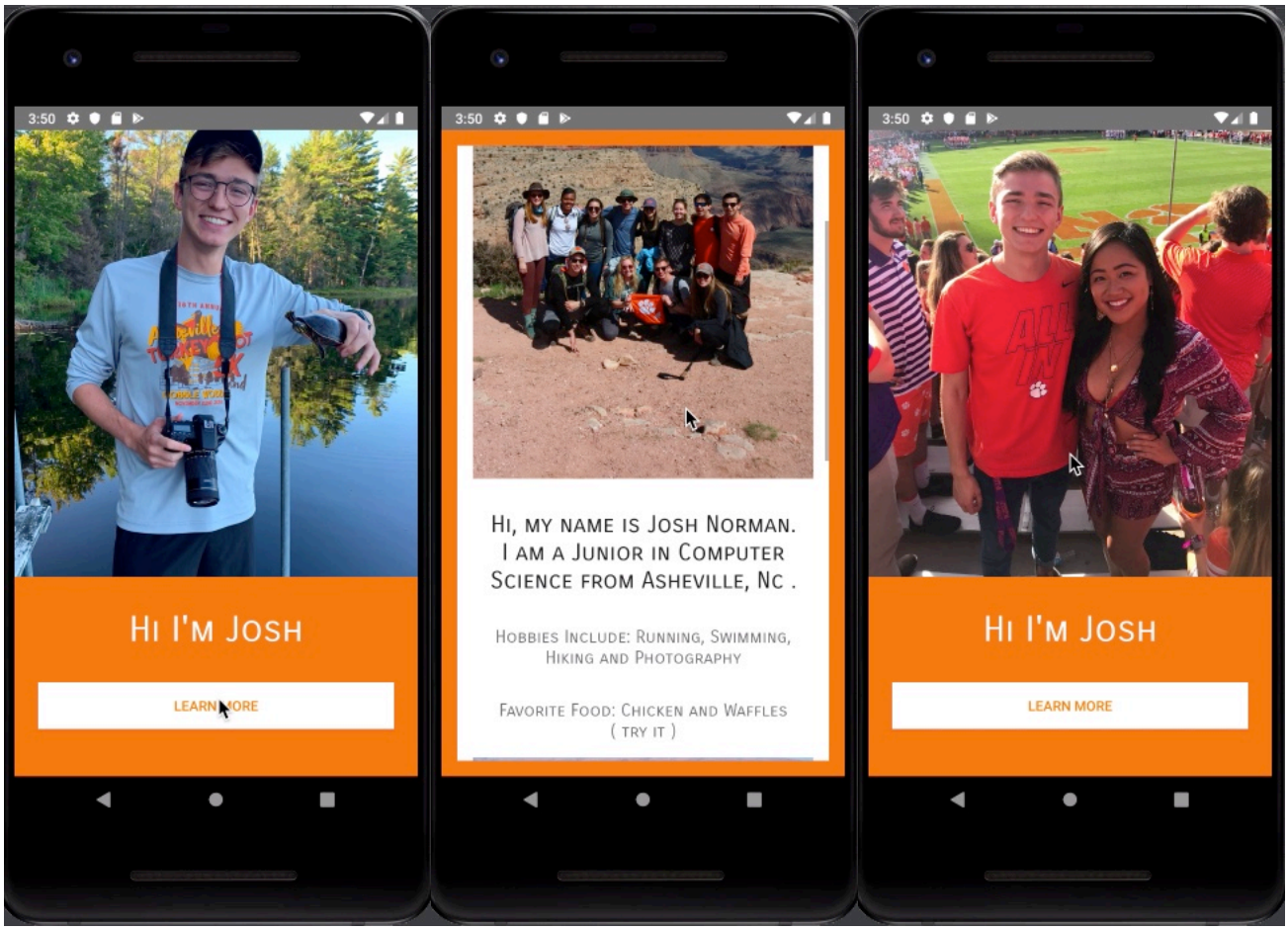
- On launch, your app will display:
    - A picture of you
    - Text that says, "Hello World, I'm *name*" (where name is your preferred first name)
    - A button that says "Learn More"
- If a user taps your picture, the image is replaced with another image. There should be a set of at least 3 images (including the one that is displayed on initial startup). Use a `switch` statement to accomplish this in your activity.
- Tapping "Learn More" will bring up a new view that contains (via scrolling vertically)
    - At least two additional pictures in the same view
    - Biographical text including:
        - Your name and hometown
        - Favorite programming language (and one sentence why)
        - Hobbies
        - Favorite food
        - One fun fact

        As a good programming practice, each one of the above strings should be stored as a separate string in `strings.xml`
- You also need to create a unique app icon that uses your first and last initials using Image Asset Studio (or just have fun with it!)

Extra Credit:

- A button in the interface that redirects to links to either the graduate student's official LinkedIn page, or a personal home page. This redirection should occur with the Android on-board browser, Chrome.
- A drop-down menu in the Main Activity that allows the user to switch some attribute related to the text on the screen (e.g., italics, color, font family). This drop-down should dynamically change what's on the screen.

Sample Screenshots:



Note, various choices related to graphical design (e.g., font selection, color choices, and layouts) are up to you. Just make sure you leverage good design practices as discussed in class.

Additional References:
https://zyBooks.com
https://developer.android.com/studio/install
https://developer.android.com/studio/projects/create-project

Hints:

The following links may provide some helpful guidance in trouble-shooting as you work through your app design

- https://developer.android.com/reference/android/widget/ImageView.ScaleType
- Handling special characters at: https://developer.android.com/guide/topics/resources/string-resource