Assignment 1: First Android App

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

Previous steps:

1. Install and set up the **Android Studio** IDE on your computer (see Appendix A).
2. If you are going to use a connected device to run/test/debug your application, you must enable the developer options and debugging on your phone (see Appendix B).
3. If you are going to use a virtual device, you must configure the AVD on Android Studio (see Appendix C).

Instructions

In this assignment, you are going to apply your creativity skills to develop your (possibly) first Android app with Android Studio. Four different project ideas are presented. You are very welcome to choose any of those or create your own. You must only submit one application. Your app must meet the following minimum requirements (whether you choose one of the suggested apps or you define your own):

- One activity (one screen)
- A minimum of 4 widgets
- A minimum of two events (e.g. respond to a button click, response to a switch selection).

Suggestions

Tip calculator: User types in how much money they spent and chooses a percent to tip. The app outputs how much money corresponds to the tip.

Discounts calculator: User types in the original price of the item and the discount to be applied. The app outputs the final price of the item before and after applying the discount.

Rock-Paper-Scissors: The user and computer each pick an option: Rock, Paper, or Scissors. The computer's choice is made randomly. The rules are the following: Paper beats Rock; Scissors beat Paper; and Rock beats Scissors. The app should keep track of the user player's score against the computer over time.

Scrabble-mini: The computer thinks of a word and displays the letters on the screen on a random order (but never the correct order). The user must guess the word by putting the letters in the right order.

****Make your own:** they only restriction is that it should meet the minimum requirements presented before.

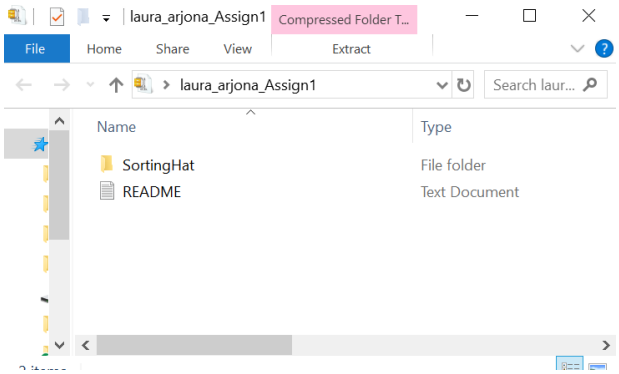
Submission

Submit your assignment using Canvas, under the Assignment section. The due date is shown in Canvas. You must submit a **.zip** file, containing the following items:

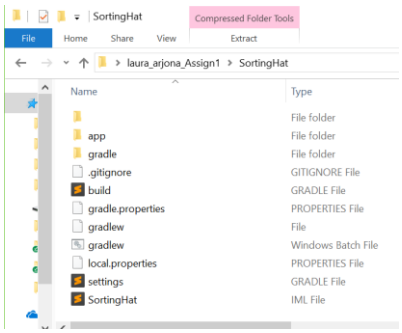
- Android Studio Project folder
- README.txt file (see instructions below)
- video file with a demo (optional for this assignment)

The file named README.txt should contain 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:

```
Laura Arjona <arjonal@uw.edu>
Sorting Hat - This app displays the sorting hat and a picture of the user. It returns the house
where the user belongs.
```



Example of content of .zip archive



Example of content of Android Project "SortingHat"

Grading

Your submission will be graded by building and running it and evaluating its functionality. Your code will not be graded on style, but we still encourage you to follow good overall coding style for your own good (see Additional resources for reference).

Additional resources

* Kotlin for android style guide with Kotlin: https://developer.android.com/kotlin/style-guide
* Resources to learn Kotlin:
  - ebook (available at UW library)
    Kotlin programming: the Big Nerd Ranch guide. Josh Skeen, author. David Greenhalgh, author. Big Nerd Ranch (Firm), publisher. 2018 Atlanta, GA : Big Nerd Ranch

  - Print Book (available at UW library)
    Kotlin in action. Dmitry Jemerov, author. Svetlana Isakova, author.
    2017 Shelter Island, NY: Manning Publications Co.

  - Full Kotlin reference: https://kotlinlang.org/docs/kotlin-docs.pdf

* Resources to learn Android with Kotlin
  - ebook (available at UW library).
    Android development with Kotlin : learn Android applicationdevelopment with the extensive features of Kotlin. Marcin Moskala, author. Igor Wojda, author.
    2017 Birmingham, UK : Packt Publishing

  - Android Development with Kotlin
    https://www.packtpub.com/application-development/android-development-kotlin

Appendix A: Download, Install and set up Android Studio

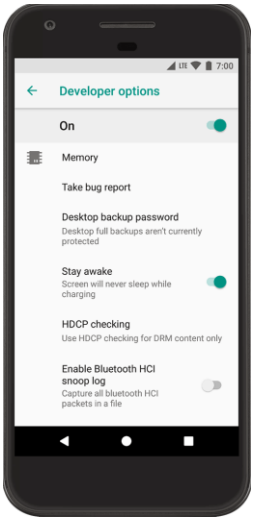
In order to develop Android apps with your computer, you will need to install two pieces of software.

1. **Oracle Java Development Kit ("JDK")**
   To run Java code in your app, you will need the Oracle Java compiler and libraries on your system. These are collectively called the Java Development Kit or "JDK" for short. Download the JDK from the following address:
   https://www.oracle.com/technetwork/java/javase/downloads/index.html
   A common mistake here is to accidentally download the Java Runtime Environment, or "JRE", instead of JDK. The JRE is not the right download; JRE enables you to run Java programs but not compile or develop your own. Be careful to download JDK and not JRE.). Once the JDK installer is done downloading, run it to install JDK on your system. You can use all of the default settings.

2. **Android Studio SDK** .
   Download Android studio for your own platform (works with Windows, Linux, and Mac) from the following link: https://developer.android.com/studio/#downloads.
   Please download the last version, so that we all have the same version and we can easily build and run your apps. To install in your own platform, follow the guide from android developers website https://developer.android.com/studio/install

Appendix B: Enable developer options and debugging

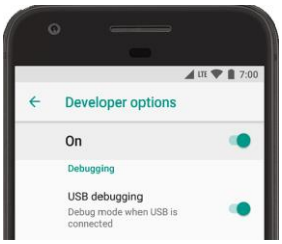
On Android 4.2 and higher (but lower than version 8.0), you must enable this screen as follows:



1. Open the **Settings** app.
2. (Only on Android 8.0 or higher) Select **System**.
3. Scroll to the bottom and select **About phone**.
4. Scroll to the bottom and tap **Build number** 7 times.
5. Return to the previous screen to find **Developer options** near the bottom.

At the top of the **Developer options** screen, you can toggle the options on and off (figure 1). You probably want to keep this on. When off, most options are disabled except those that don't require communication between the device and your development computer.

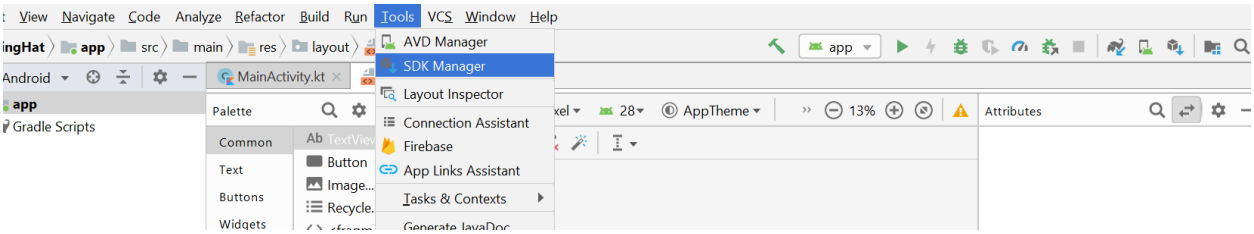
Next, you should scroll down a little and enable **USB debugging**. This allows Android Studio and other SDK tools to recognize your device when connected via USB, so you can use the debugger and other tools.



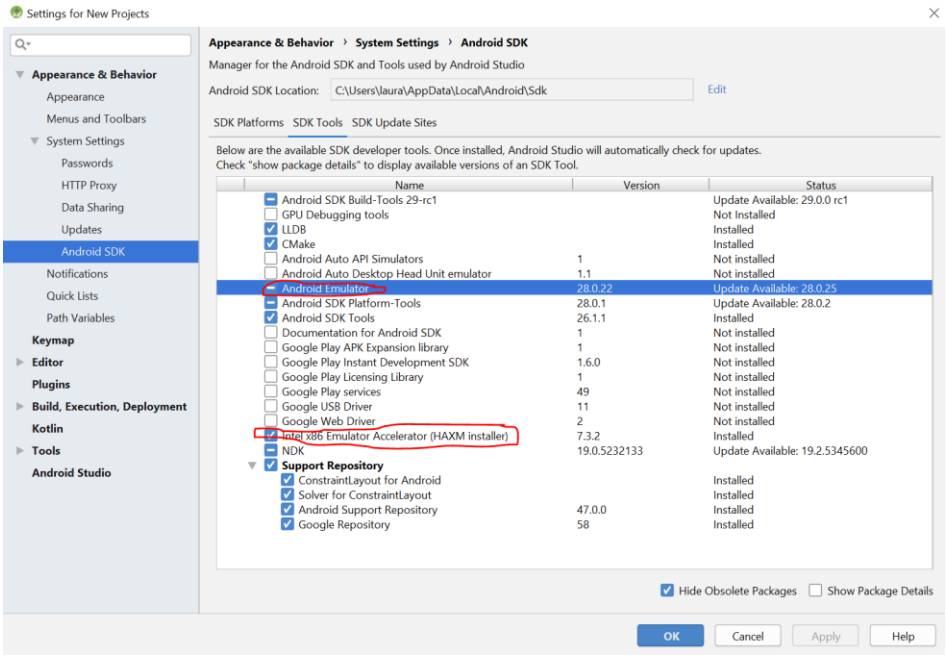
Appendix C: Setting a Virtual Device

1. Create a virtual device.
   https://developer.android.com/studio/run/managing-avds

2. Open the SDK Manager and



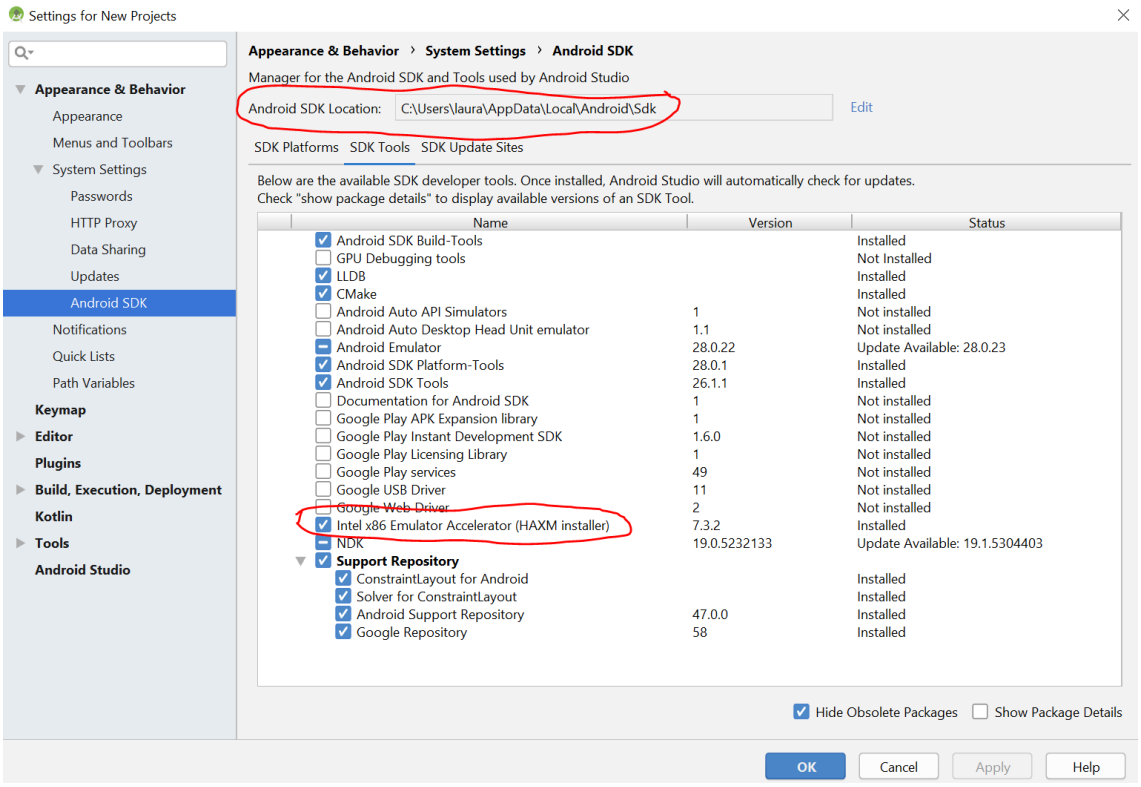
install 1) Android Emulator, and 2) the Intel x86 Emulator Accelerator (for Windows)

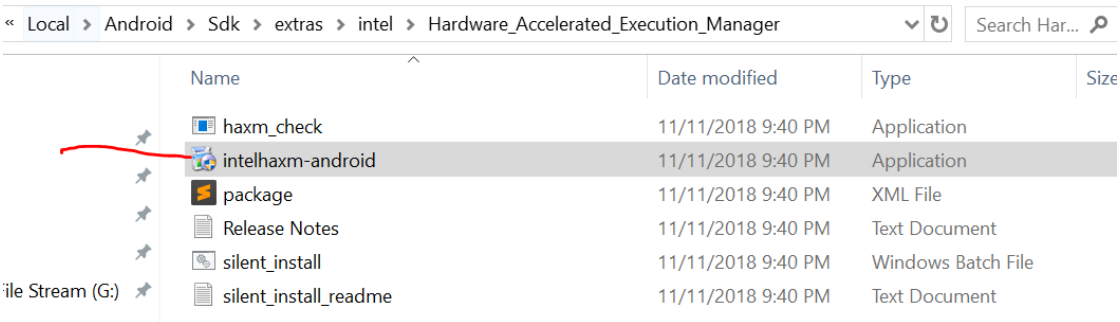


For windows users:
When running your app with the virtual device, you might encounter the following error

```
emulator: ERROR: x86 emulation currently requires hardware acceleration!
Please ensure Intel HAXM is properly installed and usable.
CPU acceleration status: HAX kernel module is not installed!
```

This is how I solved it. The Android SDK Manager **does not actually install HAXM**, it just downloads it. In the top of the Android SDK manager window, you can find where the installer is located on your PC. Please open the subfolder extras\intel\Hardware_Accelerated_Execution_Manager, and run the installer manually: *intelhaxm-android.exe* (just click on this file)
In my laptop, this file is in:
C:\Users\laura\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager





More information about configuring the emulator can be found in

https://developer.android.com/studio/run/emulator