CMPS 312 Mobile Application Development

LAB 1: Setting Up the Android Development Environment

Objective

- Set up your Android development environment
- Create and debug a simple Android application.

Note: Before starting the lab, please create a Deliverables directory on your computer desktop. This directory will hold various work products that you will turn in to demonstrate that you have completed the lab.

PART A – Downloading and Installing Android Studio

1. Download Android Studio from the Android developer's website
   a. https://developer.android.com/studio/
   b. Detailed tutorial on android studio installation can be found at https://developer.android.com/studio/install
2. While the Android Studio download completes, verify which version of the JDK you have: open a command line and type javac -version. If the JDK is not available or the version is lower than 1.8, download the Java SE Development Kit 8.
   a. You can use the below link if you don't know how to install the JDK https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/
3. To install Android Studio on **Windows**, proceed as follows:
   a. Launch the .exe file you downloaded.
   b. Follow the setup wizard to install Android Studio and any necessary SDK tools. On some Windows systems, the launcher script does not find where the JDK is installed. If you encounter this problem, you need to set an environment variable indicating the correct location as follows:

   Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable JAVA_HOME that points to your JDK folder, for example, C:\Program Files\Java\jdk1.8.0_77.

4. To install Android Studio on your **Mac**, proceed as follows:
    a. Launch the Android Studio .**DMG** file.
    b. Drag and drop Android Studio into the Applications folder, then launch Android Studio.
    c. Select whether you want to import previous Android Studio settings, then click **OK**.
    d. The Android Studio Setup Wizard guides you though the rest of the setup, which includes downloading Android SDK components that are required for development.

5. Configure SDK Manager OR if you already opened the application, then go to Tools➔ SDK Manager => Make sure the following SDK tools are installed.

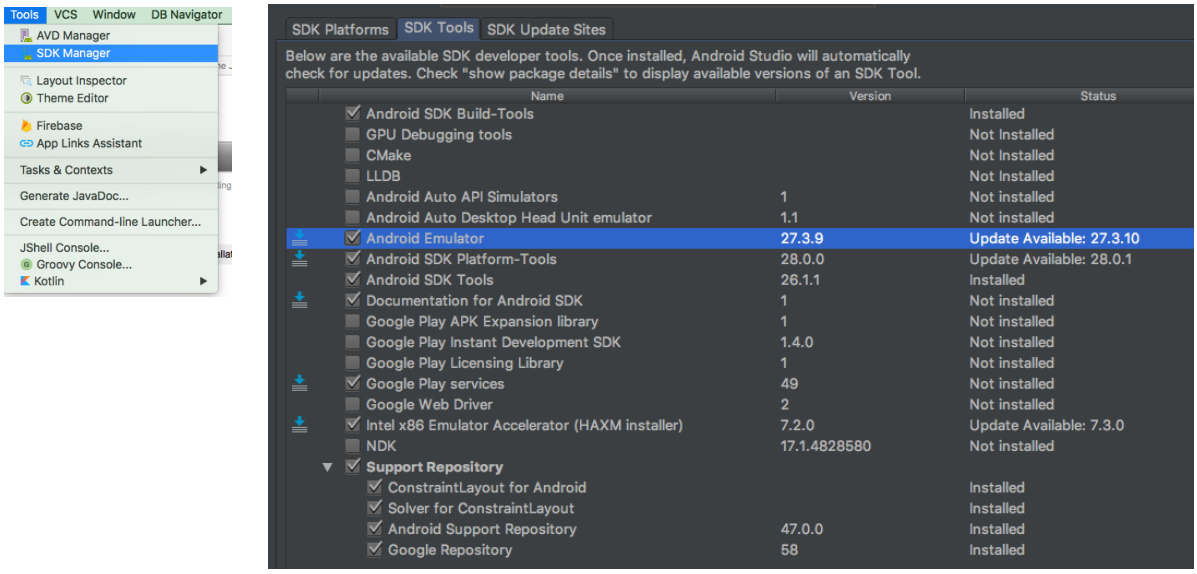
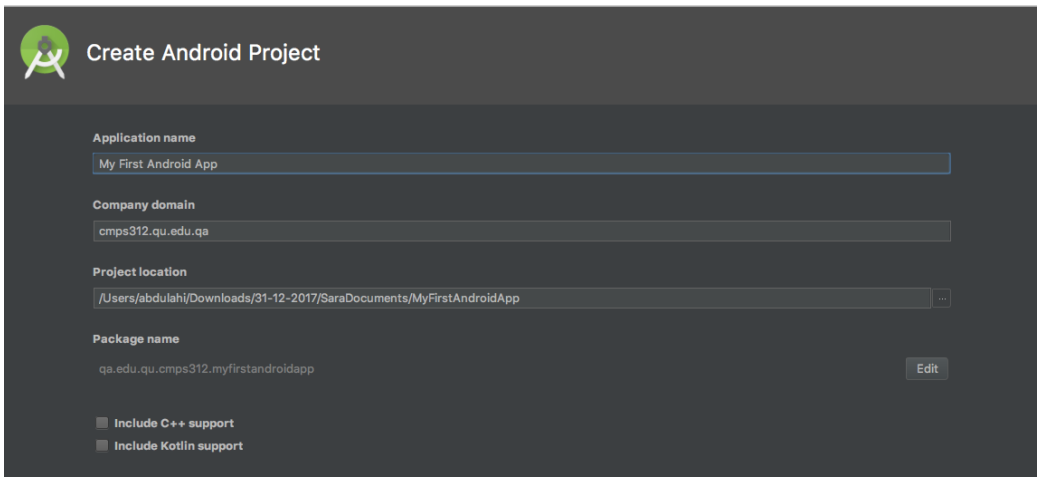


Figure 1 SDK Manager

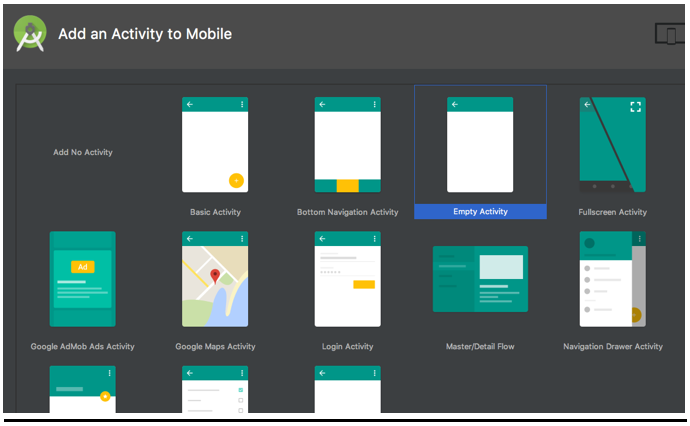
PART B: CREATING A SIMPLE "HELLO WORLD" APPLICATION

CONFIGURING THE PROJECT

1. Open Android Studio Application
2. Start New Android Studio Project
3. Make the application name: "My First Android App"
4. Domain Name: "cmps312.qu.edu.qa



5. Choose the (Phone and Tablets) and Minimum SDK 4.2 (Click on Help Me)
6. In the Add Activity to Mobile Menu Select **Empty Activity**
7. Finish



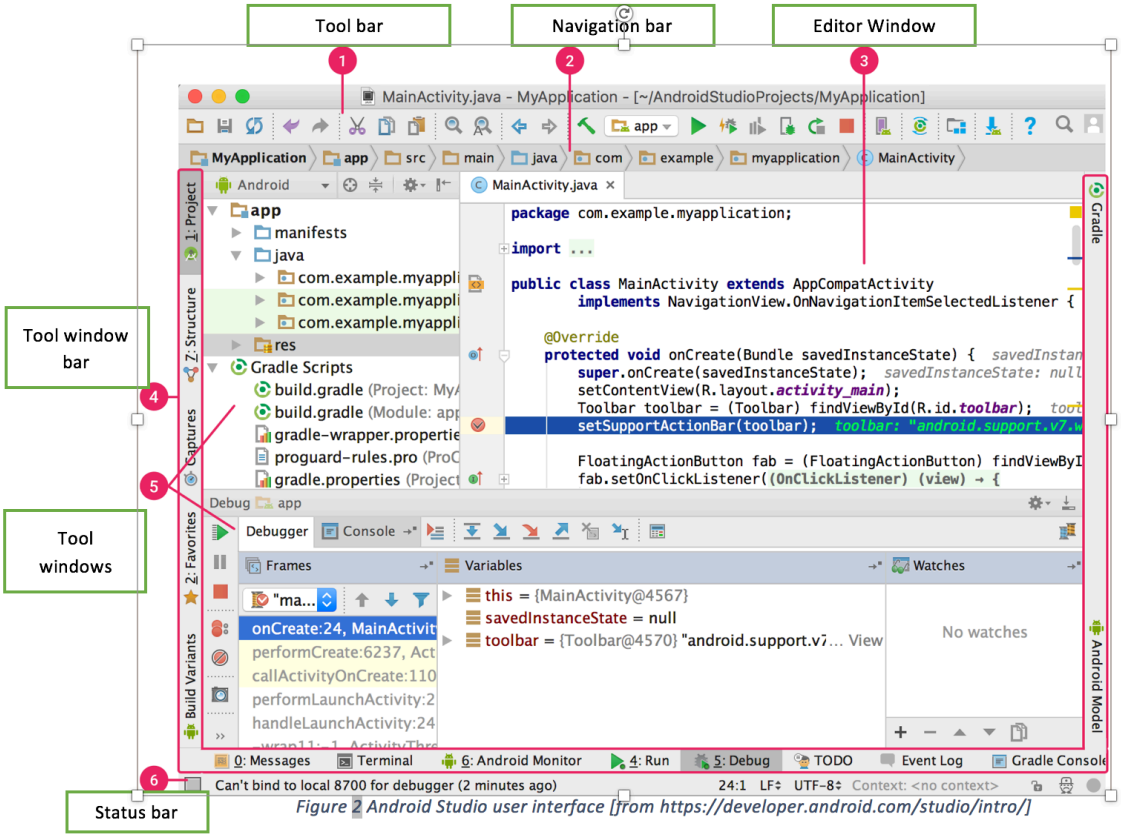


Figure 2 Android Studio user interface [from https://developer.android.com/studio/intro/]

Figure 2 Android Studio user interface [from https://developer.android.com/studio/intro/]

RUNNING THE PROJECT

A. Using Real Physical Device

1. Plug the device through USB port (Windows automatically downloads your drivers)
2. If window did not download your device's driver then go to Google "OEM Drivers from android developers" and follow the instructions)
3. Enable the developer's option on your physical phone by :
    i. Go to Settings => "About Your device" => "Build Number "then press it Seven Times. This will enable the developer's option
    ii. Search for the developer's Options inside the settings
    iii. Tick the USB Debugging Enable box

B. Using Emulator

1. Click on the AVD Manager Button on the top right corner on your android studio
2. At the bottom of the menu, click on “Create Virtual Device”
3. You will get three different choices
    i. You can define new Hardware Profile – Look on the internet
    ii. You can Import Profile from someone else’s AVD
    iii. Use the predefined emulators
    iv. Combination (Best)

Run the Application by pressing on the Run button on the top corner

Using System Logs

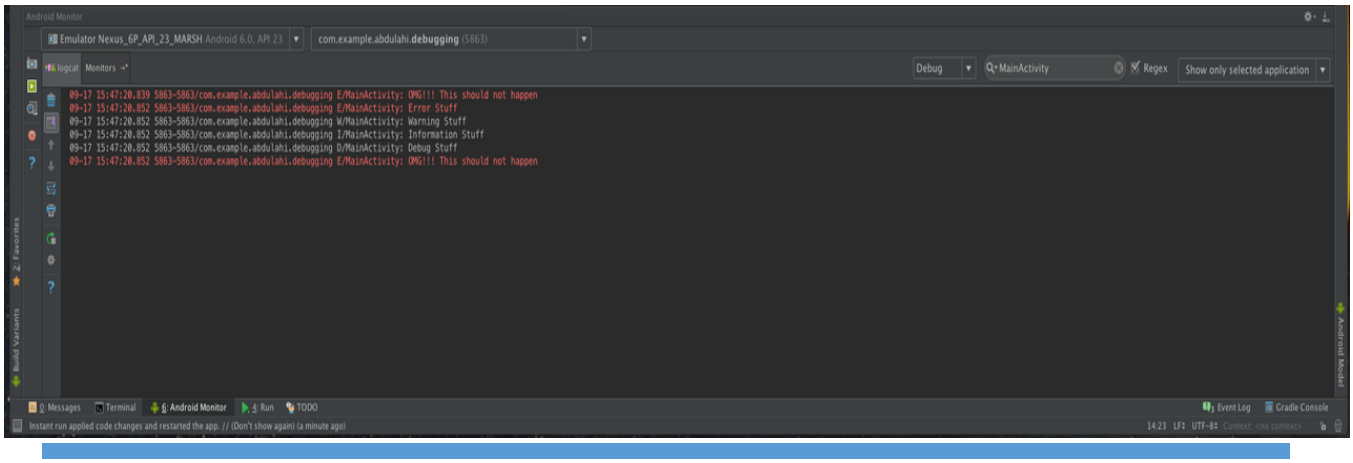
i. Log.v(); // Verbose (If you want to log everything in your app)
ii. Log.d(); // Debug (You want to view some variables values)
iii. Log.i(); // Info (Successfully downloaded the file/ connected)
iv. Log.w(); // Warning (When unexpected thing happened)
v. Log.e(); // Error (Something bad happened ლ(ಠ_ಠლ)

Extra

vi. Log.wtf()// (What a Terrible Failure) When something extremely bad happened ヽ(ಠ_ಠ)ノ

Add the above Logs to the Hello World Application

i. Declare private static final String TAG = "**MyMainActivity**";
ii. Log.”**X**”(**TAG**, "**Message**”); (Replace the X by the above Log type)
iii. Filter the different Log Messages by using the Android Monitor window

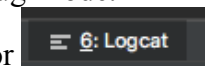


Capturing Screenshots-

i. Run your App in Debug Mode.
ii. Click on Logcat 

at the bottom.
iii. Click Screen Capture  on the left.
iv. Optional: To add a device frame around your screenshot, click Frame screenshot.
v. Click Save.

Capturing Video

i. Run your App in Debug Mode.
ii. Click Android Monitor 

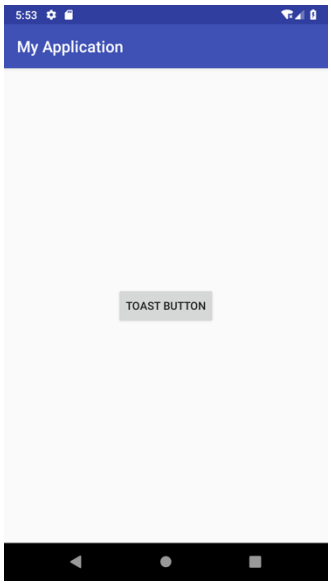
at the bottom.
iii. Click Screen Record  on the left.
iv. Click Start Recording.
v. Interact with your app.
vi. Click Stop Recording.
vii. Enter a file name for the recording and click **OK**.

Listening for Events

i. Open the activity_main.xml file that is under the layout folder. The path is shown below

ii. Remove the hello world text box and add a button
iii. Give the button an ID -> btn_toast
iv. Add a click listener to the button.
v. Call the method that handles the click **displayMessage**
vi. Add a toast message inside the **displayMessage** Method.

```
public void displayMessage(View view) {
    Toast.makeText( context: this,  text: "Welcome to CMPS312",
            Toast.LENGTH_SHORT).show();
}
```



PART E-

EXCERCISE

Develop an app to convert between temperature systems, according to the following equations

$$F=C \times 1.8+32$$

$$C=(F-32)/1.8$$