Practice Lab – 1
Topic: Android Studio, First App, Run, Logcat, Debug, Samples
Date: Jan 17, 2017

Prelab Activities:

1. Install Java 7 (or Java 8): http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Practice Lab Activities:

<table>
<thead>
<tr>
<th>#</th>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check Android SDK and JDK location: On the “Welcome to Android Studio” screen, go to Configure &gt; Project Defaults &gt; Project Structure.</td>
<td>In my PC, these are: Android SDK location: C:\Users\Nirjon\AppData\Local\Android\Sdk JDK location: C:\Program Files\Java\jdk1.8.0_66</td>
</tr>
<tr>
<td>2</td>
<td>Check Android SDK: On the “Welcome to Android Studio” screen, go to Configure &gt; SDK Manager</td>
<td>You should be able to see which Android SDK versions are installed. You can install a version that is compatible with your device.</td>
</tr>
<tr>
<td>3</td>
<td>Start a new Android Studio Project:</td>
<td>o Name the App, domain name, have a look at the path. o Select Phone and Tablet, Min SDK = API 15. (Carefully notice what it means) o Select an “Empty Activity”. We want to start from the scratch. o You may choose to rename the name of the “Activity” and “Layout” if you want.</td>
</tr>
<tr>
<td>4</td>
<td>Be familiar with various windows:</td>
<td>o ALT 0: Messages – build errors, warnings etc. o ALT 1: Project – manifests, java, and res o ALT 6: Android Monitor – logcat, Memory, CPU, GPU, Network o ALT 7: Structure – shows classes and methods in a nice way</td>
</tr>
<tr>
<td>5</td>
<td>Open (using ALT 1) and study the java file of the main activity: java &gt;</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Open (using ALT 1) and study the layout of this activity: res &gt; layout &gt; *.xml</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Open (using ALT 1) and get introduced to AndroidManifest.xml</td>
<td></td>
</tr>
</tbody>
</table>
8 Building and running the App on the Emulator:
   - Click the Play button.
   - Choose: Launch emulator; Create a new virtual device (Nexus 5, 1 GB RAM, Webcam as Cameras)
   - In case of “HAXM” error, do the following 3 extra steps:
     - Tools > Android > SDK Manager. Select Tab SDK Tools, and install Intel x86 Emulator Accelerator (HAXM installer). It will just download the installer, but will not install.
     - In your computer, go to <SDK Folder>/extras/intel/ and install HAXM.
       - If the installation fails, that means you need to enable Virtualization Technology (VT-X) in your PC. Restart PC and enter BIOS. Enable “virtualization” in the Security Tab.
       - Try again. If it fails again, it may be because you have Hyper-V (or similar virtualization software) installed. Uninstall Hyper-V from Control Panel in Windows.
       - Try again.
   - Emulator at its first launch may take several minutes to load. Wait with patience!
   - Try not to close the emulator. Running Apps on the same emulator saves time.

9 Try these changes:
   - Rename the App.
   - Message in the TextView.
   - Adding a new TextView in the xml layout
   - Change the layout to LinearLayout. Place the two TextViews vertically.

10 Create a new method, and call it from the onCreate(). Inside the method, print a message using Logcat.
   - Log.v("SOME_TAG", “your message”);

11 Debug the APP:
   - Add a variable, x and assign it different values at different lines.
   - Set some break points.
   - Attach a Debugger.
   - Switch to Debugger tab (bottom).
   - Click on the Debug icon to start debugging.
   - Use step_out, step_into etc., see values of variables, watch some variable, etc.

12 Work on an existing Sample Project (comes with SDK):

Import Android Code Sample, and Run it.
  e.g. Choose UI>ActionBarCompat-ListPopupMenu project.