

IEMS5722 Assignment 0: Android Development Basics

Submit to Blackboard by **23:59, Friday, January 24, 2020**.

# Notes:

1. You are strongly advised to go through all the steps in this assignment in order to get familiar with the Android development environment.
2. Pay attention to Section 4, where you are required to perform some tasks.
3. See the instructions at the end of this assignment, follow them to submit your files for marking.
4. Late submissions will receive **30% mark penalty**.
5. This assignment accounts for **4%** of your final grade.

# Aim

To learn the fundamentals of Android development from project creation to installation on a virtual or physical device.

# Objectives

* + Set up the Android development environment
  + Create a Hello World Android application
  + Understand the Android project structure
  + How to run an app in the emulator
  + How to deploy the app to a physical device

# Instructions

* 1. **Setting up the development environment**
     1. **Android Studio**

The Android Studio is the standard environment for developing Android app development and is the recommended option for this course.

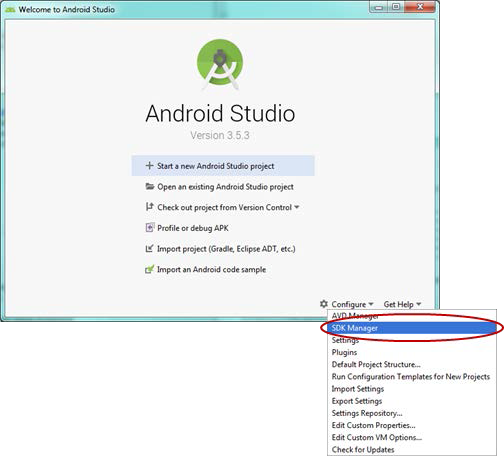
Download from <https://developer.android.com/studio>

Once you have downloaded the package, follow the online instructions to install it on your computer. When you start Android Studio for the first time, it will ask you to specify a workspace. This is where your project’s files will be saved. You can choose a default location to use or change it later if you want.

# Installing the Android SDK Packages

Android Studio does not contain all the packages needed for app development yet. These packages can be downloaded using an integrated tool called the Android SDK Manager.

In Android Studio, click **SDK Manager** in the toolbar. Its icon is highlighted in the below screenshot.



Click on Android SDK Manager to show the different packages available. By default, the Android Studio includes the current version, but you will need to download other versions if you want to build your app against older Android versions. Targeting earlier versions let you reach a larger number of users. You will also need to use the SDK manager to download later versions of the API when they are released in the future.

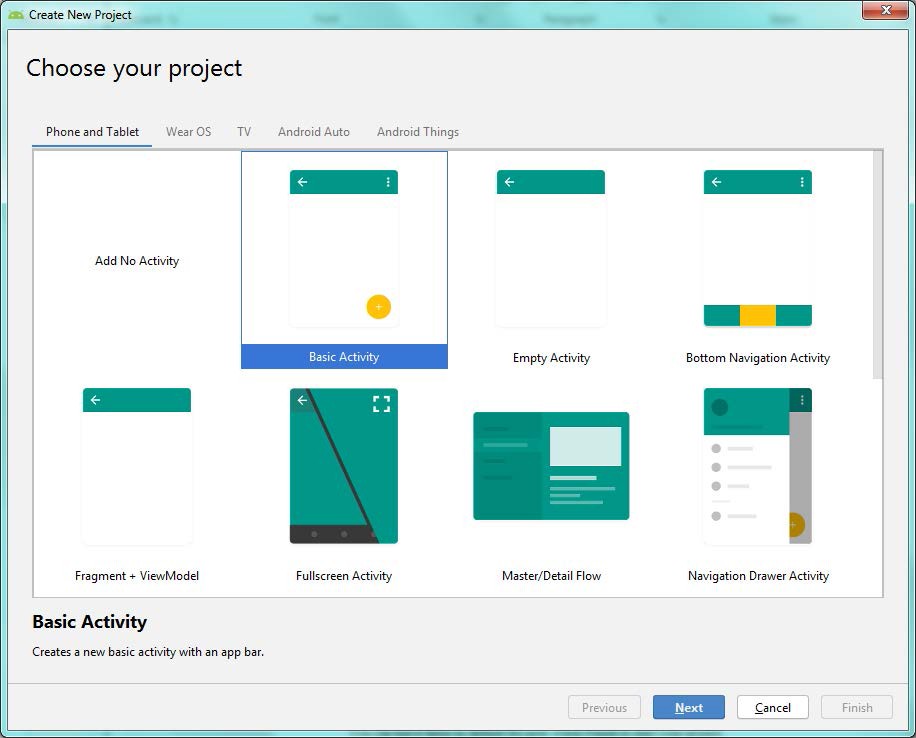
For this course, you will develop apps that support **Android 5.0 (API Level 21)** or later versions. You can download all platforms that are API 21 and later under **SDK Platform**. In addition, check that under **SDK Tools**, the Android SDK Tools, Android SDK Platform- Tools and Android SDK Build-Tools are installed.

Android Studio may need to be closed in order for Android SDK Manager to install the new packages.

# Creating your first app

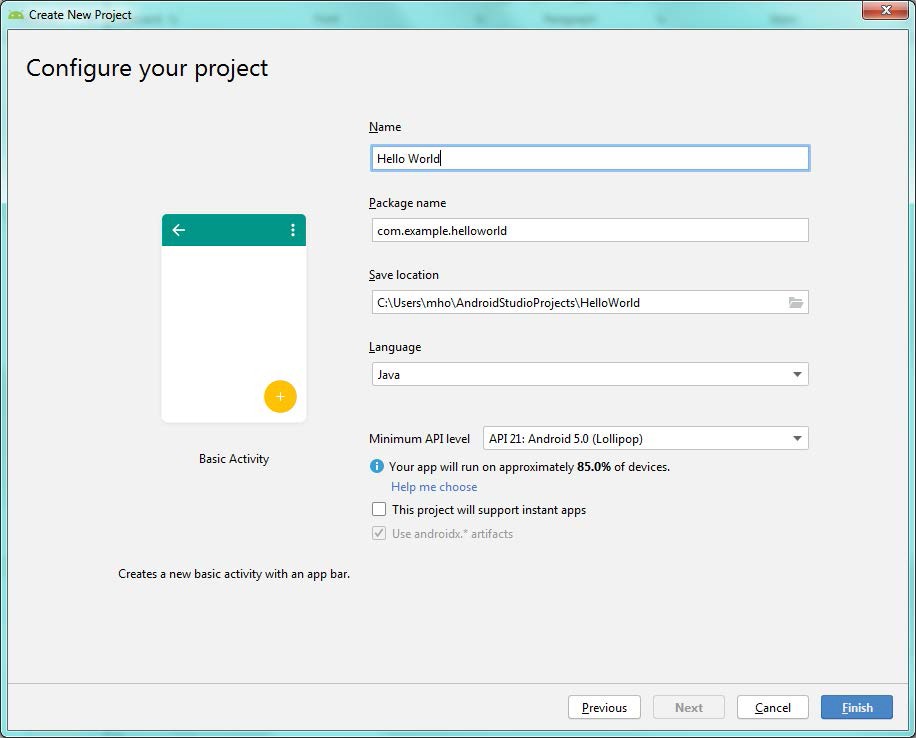
To create an Android project, select **Start a new Android Studio Project** at the welcoming screen.

We will develop for **Phone and Tablet**, and use the **Basic Activity** for this example. An example of the dialog is seen below. You are encouraged to try out the other ones to see how they work.

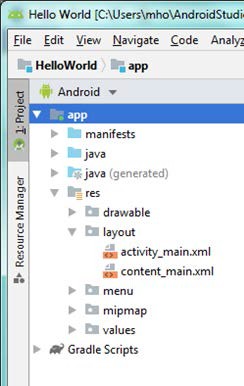


Click **Next** to set up the activity. The Configure your project dialog appears. In the Name field, enter “Hello World”. This is the name that shows up in the Android application launcher.

The app’s package name will be updated. The beginning of the package name reflects your company domain (in this case, example.com). The domain name is seem to be reversed and is following the Java package naming rules. Change the development Language to **Java**. The **Minimum API Level** is the lowest version of Android that your app supports. Lower versions target more devices, but less SDK features will be available. Set this to **API 21** for this exercise. Click **Finish** to start your project.



It may take some time for Android Studio to configure your newly created project. Once the configuration is completed, new HelloWorld project files should be visible in the Android project view panel (right).



If you choose a different option in the Android menu highlighted in red, you can visualize your project resources differently. This is useful depending on what you want to see (e.g., The Project option most closely resembles the structure from Windows Explorer)

To see what Android Studio has created, explore the project tree.

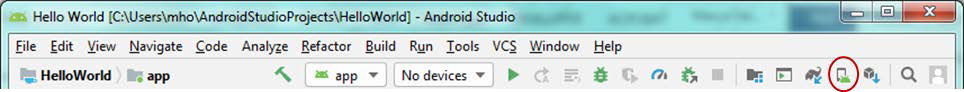
Note it has created a HelloWorld project, using the project settings we chose previously. The

*java* folder contains the Java source files, in the package *com.example.helloworld*. Likewise, the *activity\_main.xml* file has been created in the *res/layout* folder.

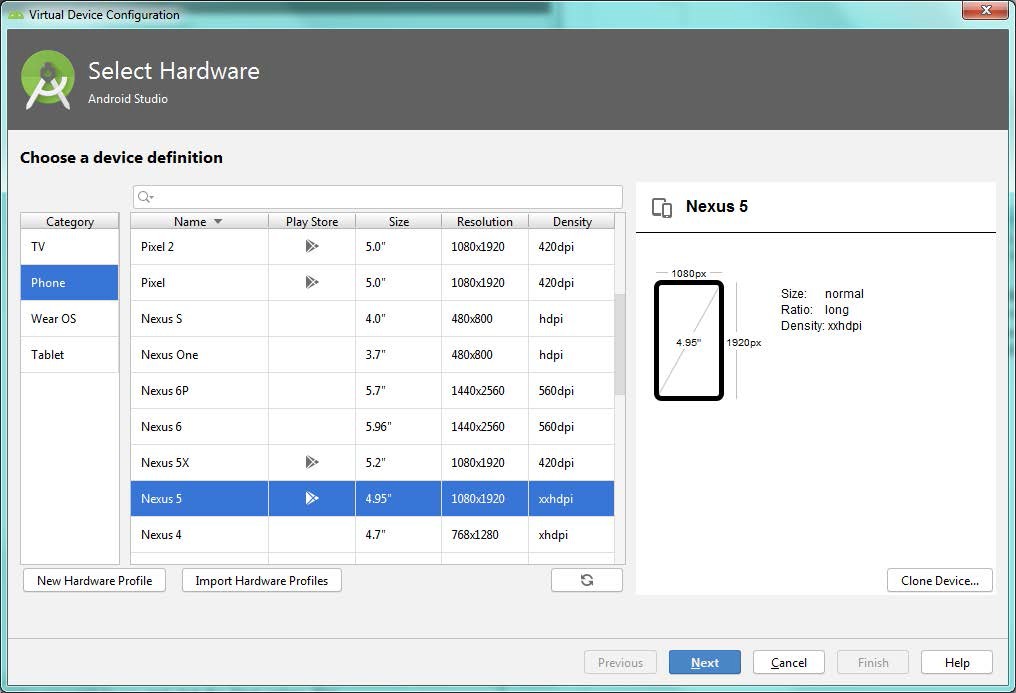
# Running your app in the emulator

* + 1. **Create an emulator profile**

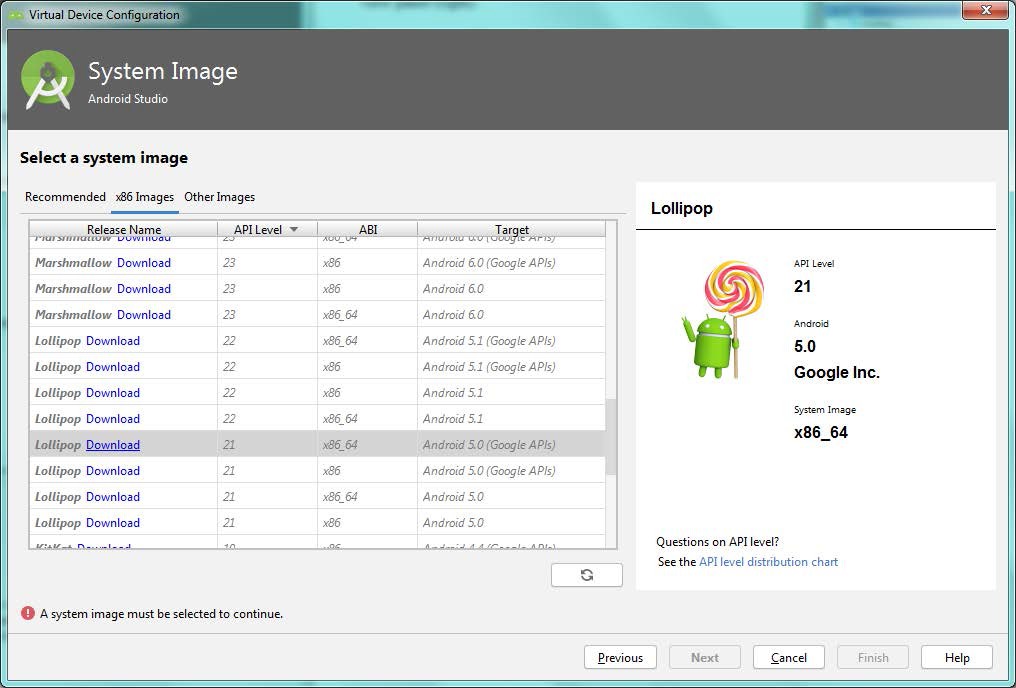
In Android Studio, go to **Android Virtual Device (AVD) Manager** (button highlighted below). From here you can create, edit and delete your emulator profiles. Click on the Create Virtual Device button to define a new emulator profile.



Emulator profiles allow you to test your app with different settings so you can see how your app behaves even if you don’t have a particular device physically. A **Nexus 5** is selected below. This is sufficient for this exercise. If you want to simulate a particular device, you will need to look up the particular devices profile. Select Next to continue.



Select an Android system image for the emulator to run. Select Lollipop (API Level 21), either x86 or x86\_64, and use the one with Google APIs. You may need to find it under **x86 Images** instead of **Recommended**. Select **Download** to download the system image.



After the image is downloaded, select **Next**, leave other options as default and create the profile.

# Starting the emulator

In the **AVD Manager window**, choose the emulator you wish to run and click the Launch button (looks like a “play” icon). The launch status is not displayed in the AVD Manager window, but at the bottom of the main Android Studio window. This may be quite slow to start (similar to cold-booting an actual Android device), but you do not need to close the emulator until you are finished.

# Deploying the app

In Android Studio, choose **Run** > **Run App**. It may ask you where do you want to deploy your app to if you have more than one emulator running or if your phone is also plugged in. Choose the emulator.

If you modify your app, you can deploy it again to the emulator without having to restart the emulator. Just choose **Run** > **Run App** again.

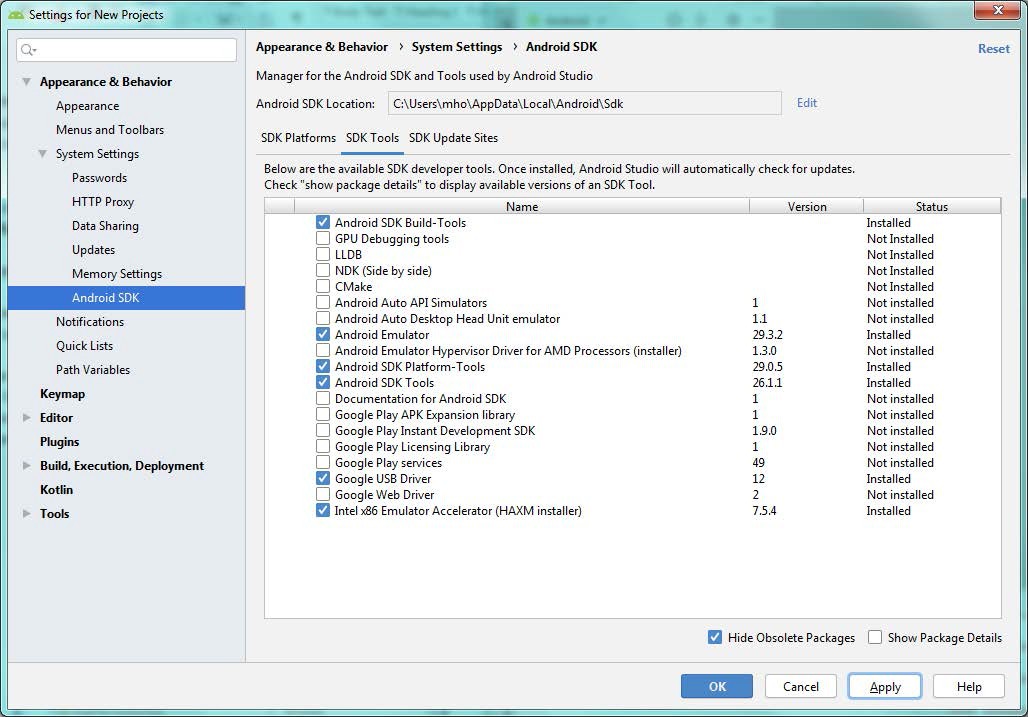
# Running your app on a physical device

* + 1. **Setting up your computer**

You will need an Android device and a USB cable to connect your device to your computer.

Install **Google USB Driver** from the **Android SDK Manager**, under **Android SDK > SDK Tools**. This will be at the near the bottom of the list. The figure shown below has the Google USB driver checked.

Note that some Android devices do not need the Google USB driver for development. The Samsung phones for example, uses the OEM driver. You will need to download Android USB driver from Samsung website. Consult your manufacturer documentation if in doubt.



# Setting up your Android device

Devices running Android 4.2+ have developer mode disabled by default. To enable it on your device, go to **Settings > About Phone > Build Number**. On some devices this may be under **Settings > About Phone > Software Information > Build Number**.

Tap on Build Number **7 times**. There should be a pop-up notification saying “You are *x* steps away from being a developer” which counts down after two taps.

If you were successful, the **Developer Options** menu should be visible under **Settings**. In the

**Developer options** menu, scroll down and enable **USB debugging**.

# Deploying your app

Connect your device to your laptop using the USB cable. On your device, you may need to switch to **MTP mode**.

In Android Studio, right click on your project and select Run > Run App. A window prompt will appear asking you where you want to deploy it to. Your physical device should appear in the list. Select your device then click to deploy.

If this is the first time you have deployed to your phone from this computer, your phone will ask you if you want to allow apps to be installed from this source. You can also save this source so it won’t ask you again next time. Accept the connection to continue installation.

The app will run when automatically when it has been installed.

# Tasks

* 1. **Modifying your app**
     1. **Changing the text**

You will notice that the app shows “Hello World” already. Look through the app files generated and find where and how the text has been displayed.

Then add in another line of text, **showing your name and student number** on the same line. **DO NOT** hardcode your text in the *content\_main.xml* file or the Java source file — use the **string resources**. As a challenge, try to put it in your text in the center of the screen.

Hint: Start from *MainActivity.java* and look at what layout it references. Then look in the layout and look at what string resource it uses.

# Changing the app icon

The app has launch icon by default. This is stored in the *res/drawable* folder, which is where graphics such as icons and images can be stored.

There are several drawable folders corresponding to different resolution devices: Low DPI, Mid DPI, High DPI and Extra High DPI. Android chooses the files most appropriate to the device that is running the app. By default, Android Studio has not created these folders but you should use these in your apps.

If the file is missing from a particular folder, it will choose the image from the next best available resolution. For example, a High resolution device will try to source its images from the *res/drawable-hdpi* folder. If it is cannot find a particular image, it will look in the *res/drawable-xhdpi* folder and scale down the resolution to fit. If not successful, it will then look in the *res/drawable-mdpi* folder and scale up the resolution. This is not particularly beautiful so you should ensure that you provide a low/med/high/extra high resolution image for your app.

Change your apps icon to be the *cuhk.png* but do not rename the file. For this exercise, you do not need to create logos of different resolutions.

# Submission

To submission your assignment, compress the whole project folder into a ZIP file and name it in the following format: **<your\_student\_id>\_a0.zip** (e.g., 1155000000\_a0.zip). Submit it to Blackboard (<https://blackboard.cuhk.edu.hk/>).

# Resources

* + Google Android Developer Guides <https://developer.android.com/guide>
  + Android Style Resources <https://developer.android.com/design/downloads/index.html>
  + Android API Reference <https://developer.android.com/reference/packages.html>
  + Where to go if you have a question (apart from Google): <https://stackoverflow.com/questions/tagged/android>