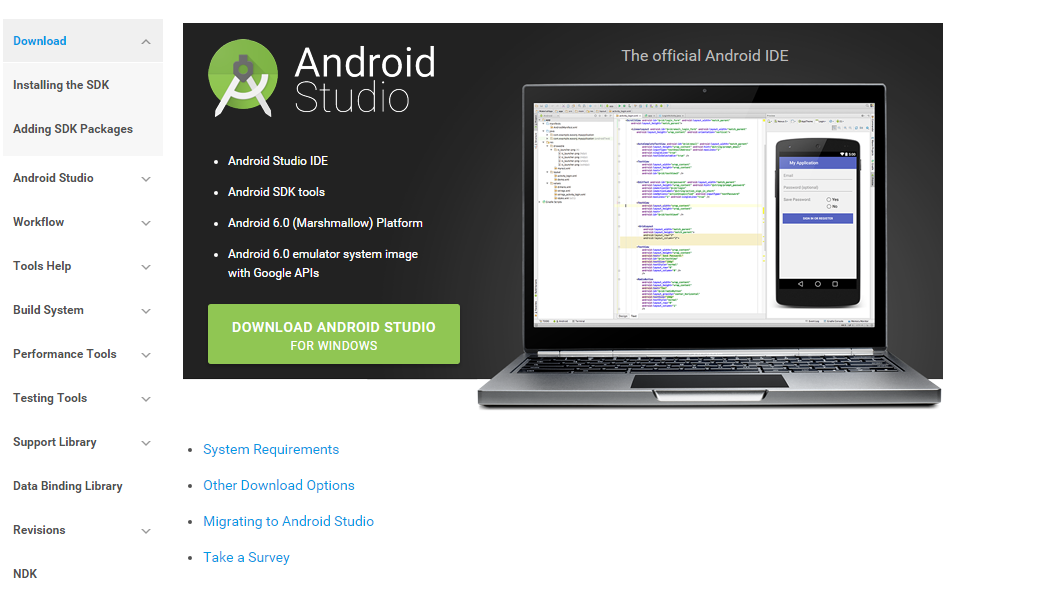
# Setup Instructions for COMP 3617

All the software for COMP 3617 course is already downloaded and available in J:\COMP\3617\2016 folder

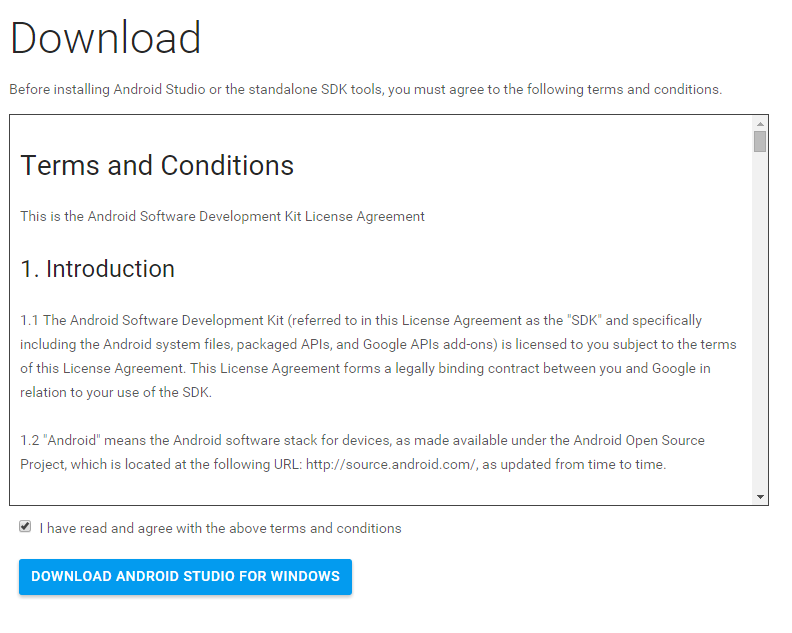
## Obtaining the Software from the Internet

If you want to obtain the software from the internet directly, here are the instructions for Android Studio and JDK 1.7

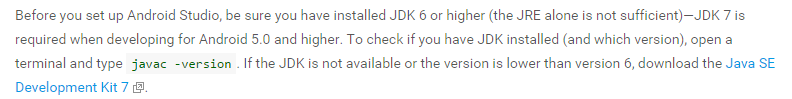
Navigate to <http://developer.android.com/sdk/index.html>



Click on Download Android Studio for Windows (If you are using Mac or Linux, you should the corresponding download link)

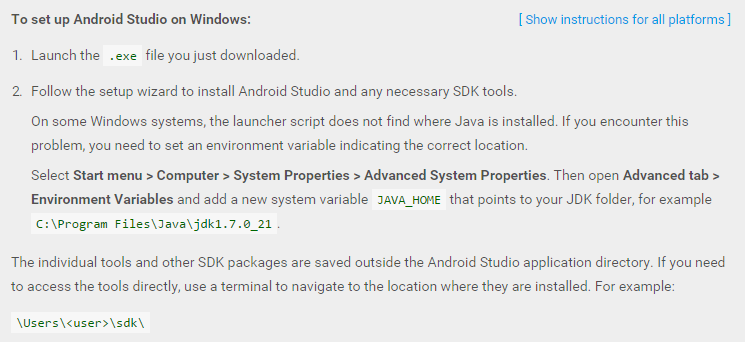


It is about a 1GB download, so depending upon your speed, it may take a while.



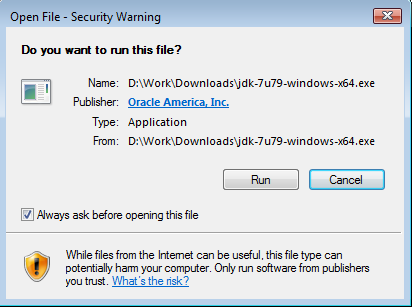
JDK 7 can be downloaded from the following location

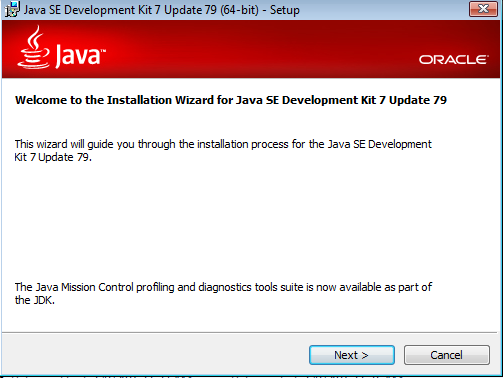
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>



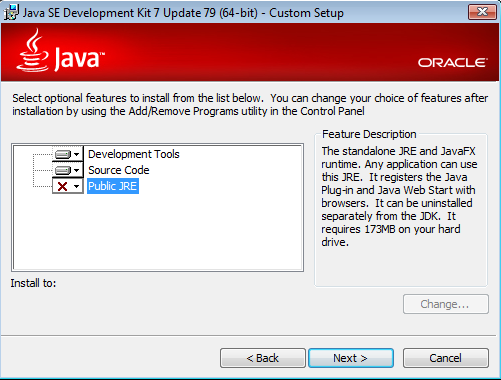
## Installation of the JDK

Click on the JDK installer

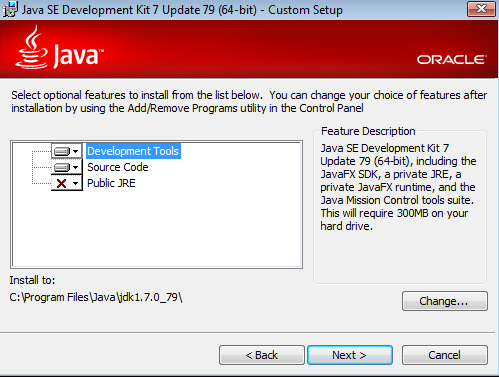


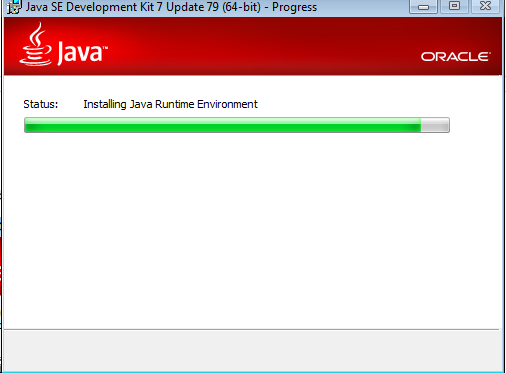


Ignore the JRE as it might impact other software that might be installed on your systems



Select the default location of the installer



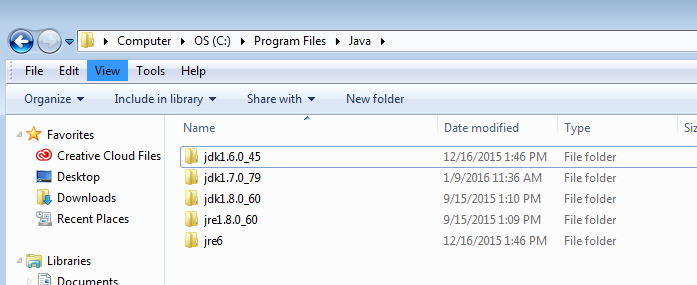


Click the Close button. There is no need to obtain the API documentation for this course.



## Multiple Java Versions on the computer

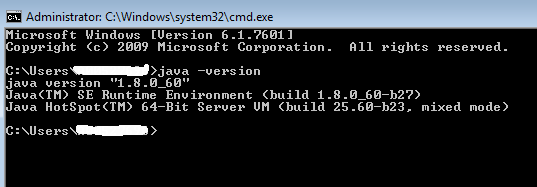
The computers in the class room have multiple installations of JDK/JRE. **Android Studio requires 1.7** and would not work with 1.6. I am not too sure about 1.8, but have heard that it works. Google’s recommendation is to use 1.7, so we will be using 1.7 in this class



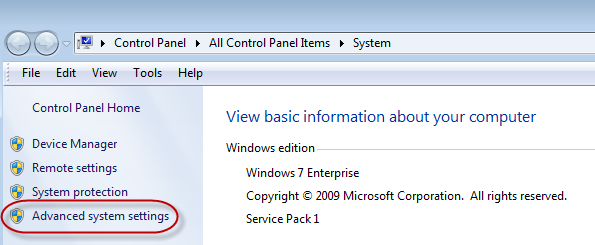
## Verifying the default JDK used by System

To determine which version of the JDK is being used by the system, open a command prompt (Windows Key + R and then type cmd)

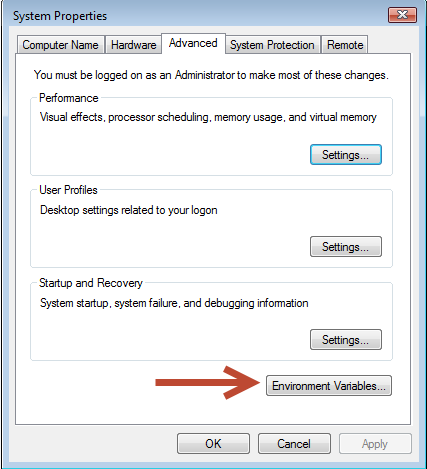
In the command prompt type ***java –version***. I see Java8 as already set up.



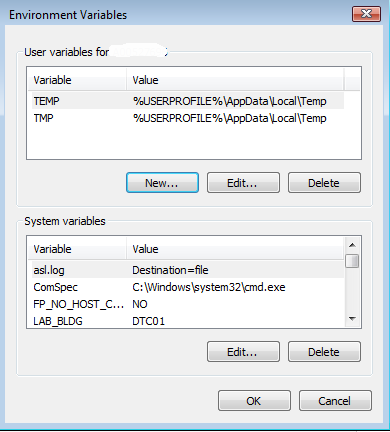
To change this to JDK 1.7, right click on Computer icon on the desktop and select Properties



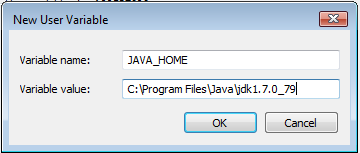
Click on Advanced System Settings



Click on Environment Variables button

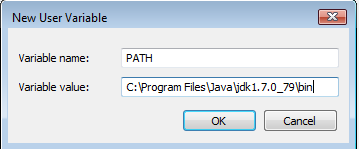


Add a new User Variable JAVA\_HOME (Click the New button on the top section).



Make sure you set the JAVA\_HOME environment variable to point to the JDK17 that you installed

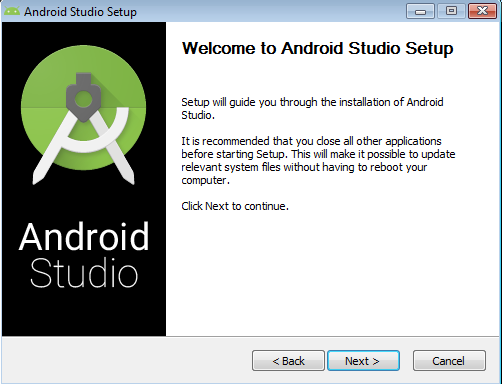
Add another User Variable for PATH. If you already have a PATH, then just prepend the Java bin path value to the existing PATH separated by “;”

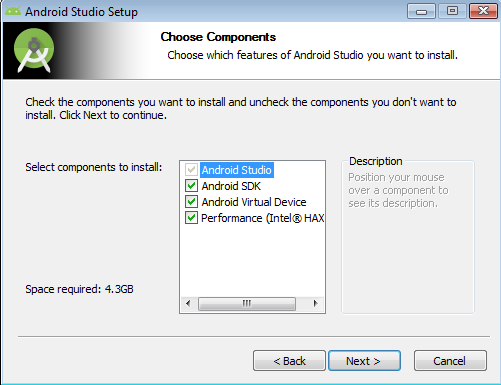


Note: This may not work if the System Path already has a JDK bin folder set up as the System Path precedes the User Path

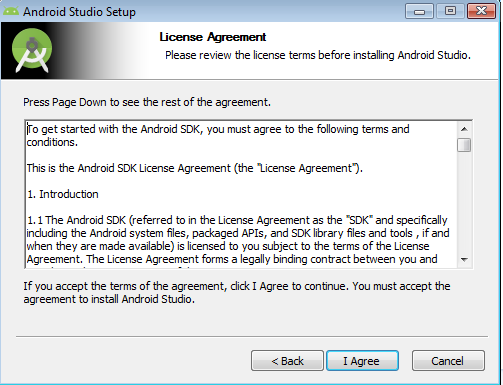
## Android Studio Installation

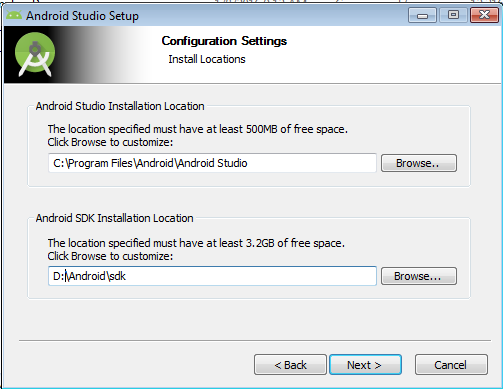
Double click on the Android Studio exe and after the security prompt, you should see the following screen



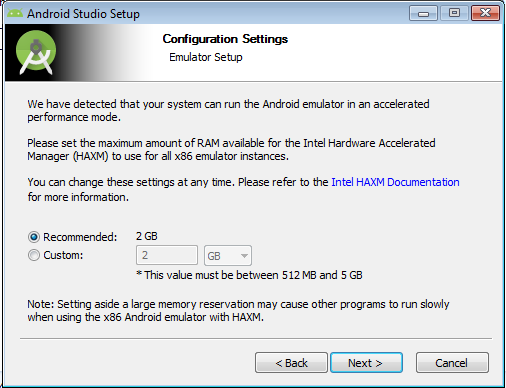


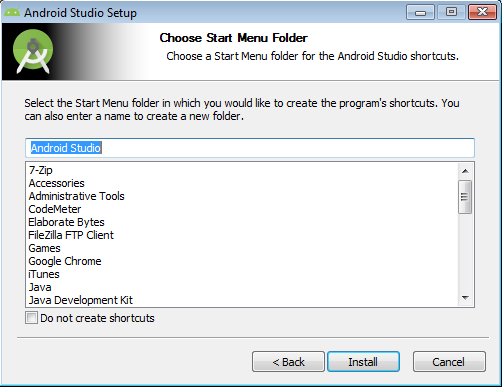
Note : Intel HAXM speeds up the emulator tremendously. Android Studio attempts to install it automatically. There are some limitations with regard to this as this feature requires the CPUs on your machine to support Intel VT-x technology. Most recent machines (from the last 5 years or so) already support this. But older machines either don’t support it or it could be that the feature is not enabled in the BIOS.

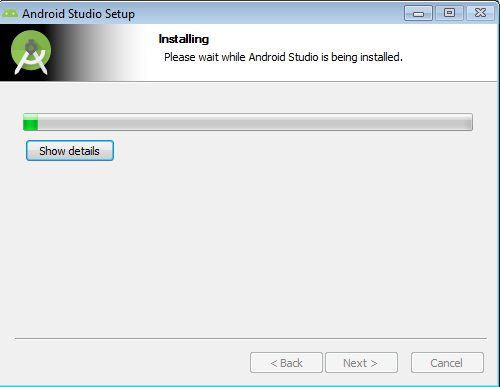


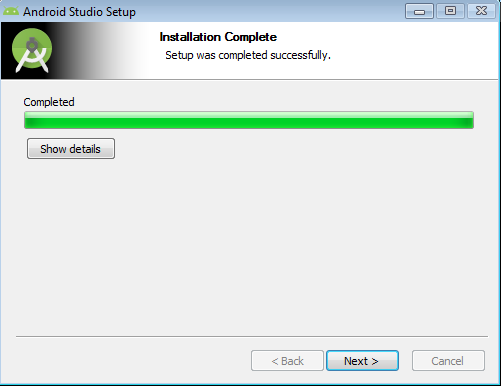


Note: If you have two drives (C and D), change the default location of the SDK to D.



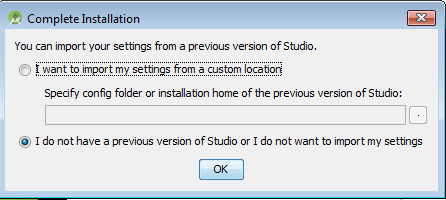


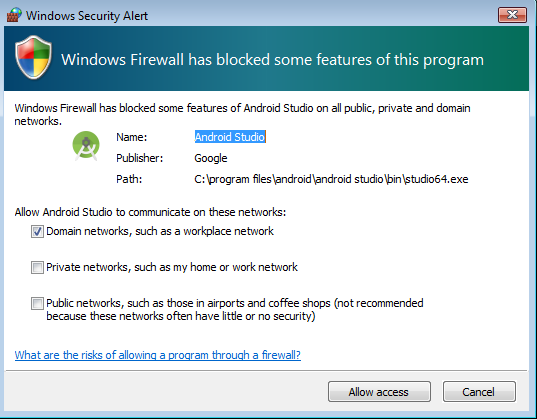




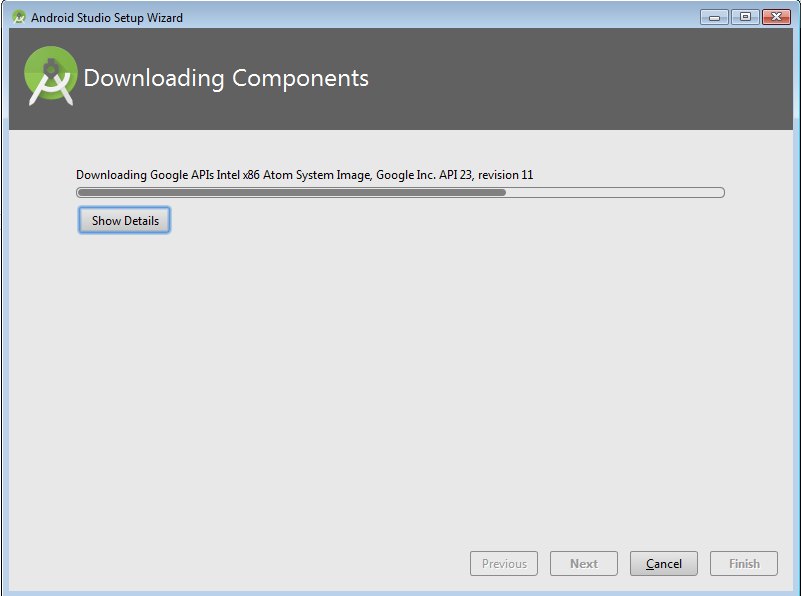


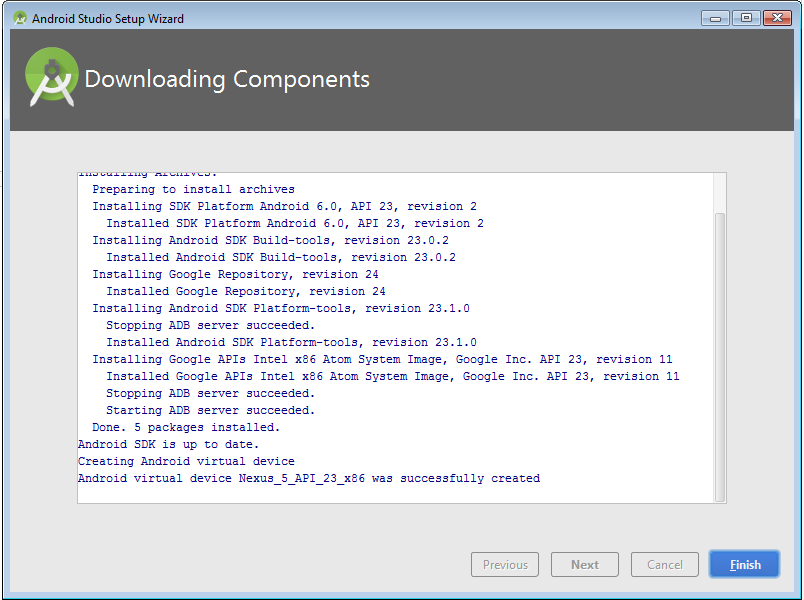
Now the Android Studio will be launched so that it can be configured

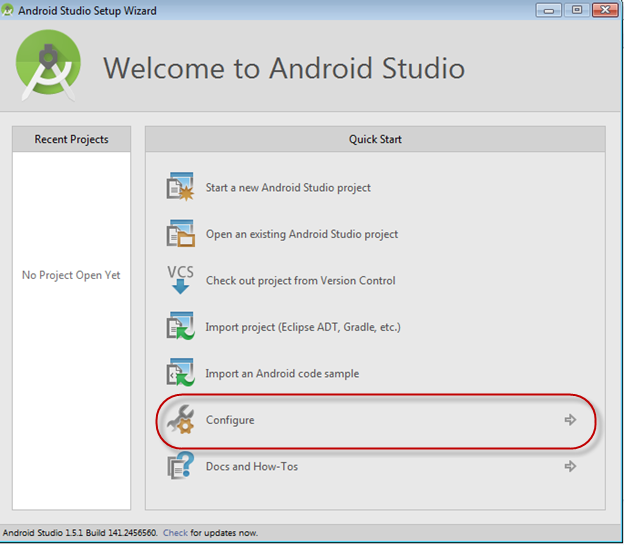




Downloads the default SDK (which is usually the latest one) automatically and creates and AVD



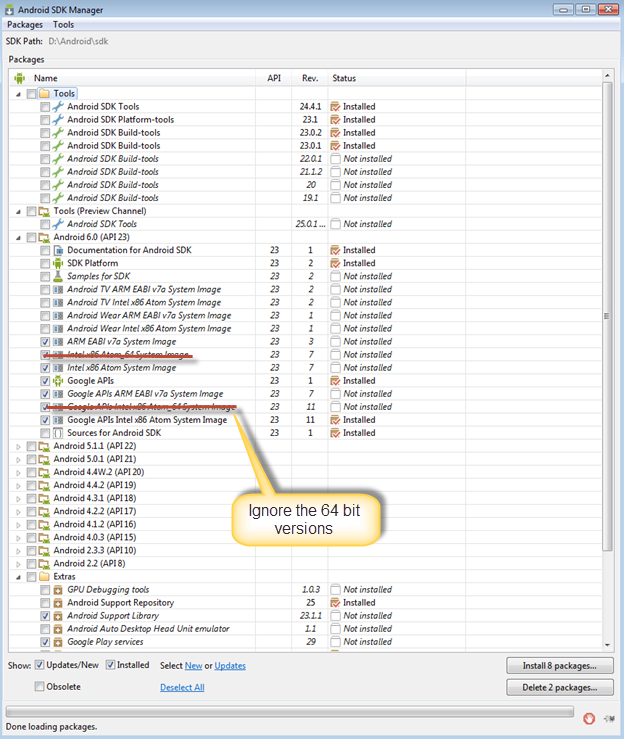




Click on Configure

This will launch the standalone tool “SDK Manager” from

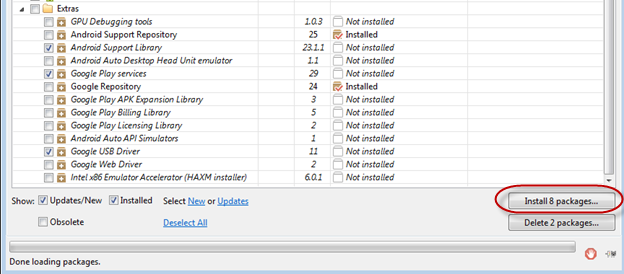
"C:\Apps\Android\sdk\SDK Manager.exe"



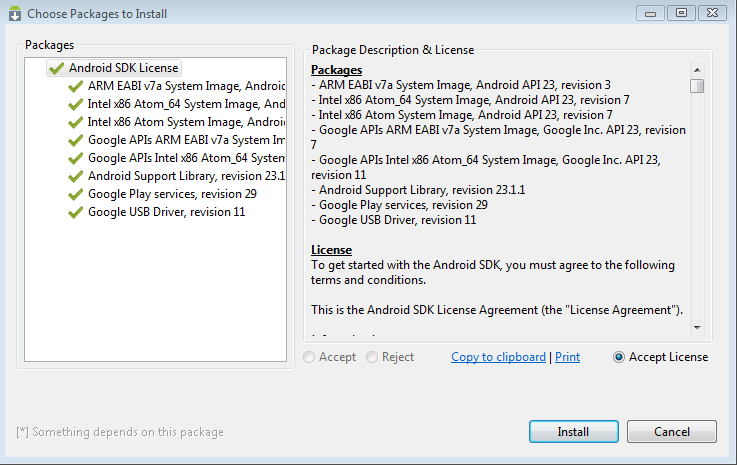
Select the following under Android 6 (API 23)

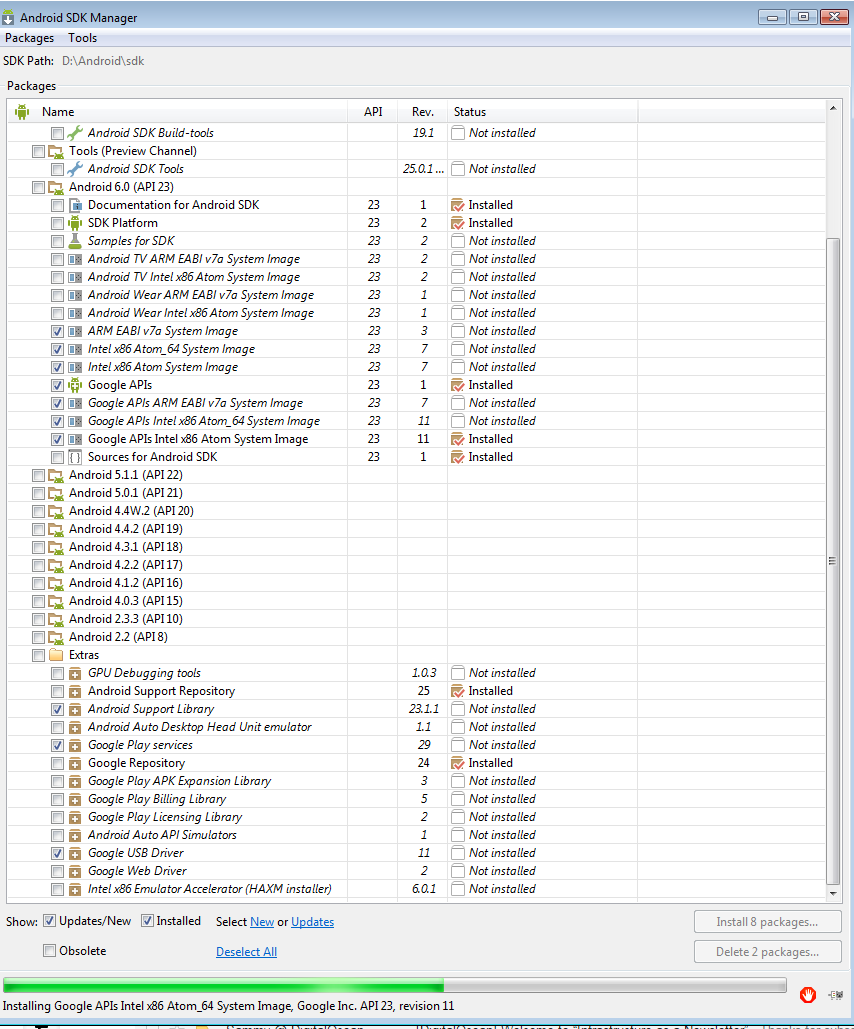
* ARM EABI v7a System Image
* Google APIs ARM EAB v7a System Image
* Intel x86 Atom System Image
* Google APIs
* Google APIs Intel x86 Atom System Image
* Ignore the x64 images

You would normally not use ARM based images as they are extremely slow but if you have issues with HAXM because your computer is an older one, an alternative is to try ARM based image.



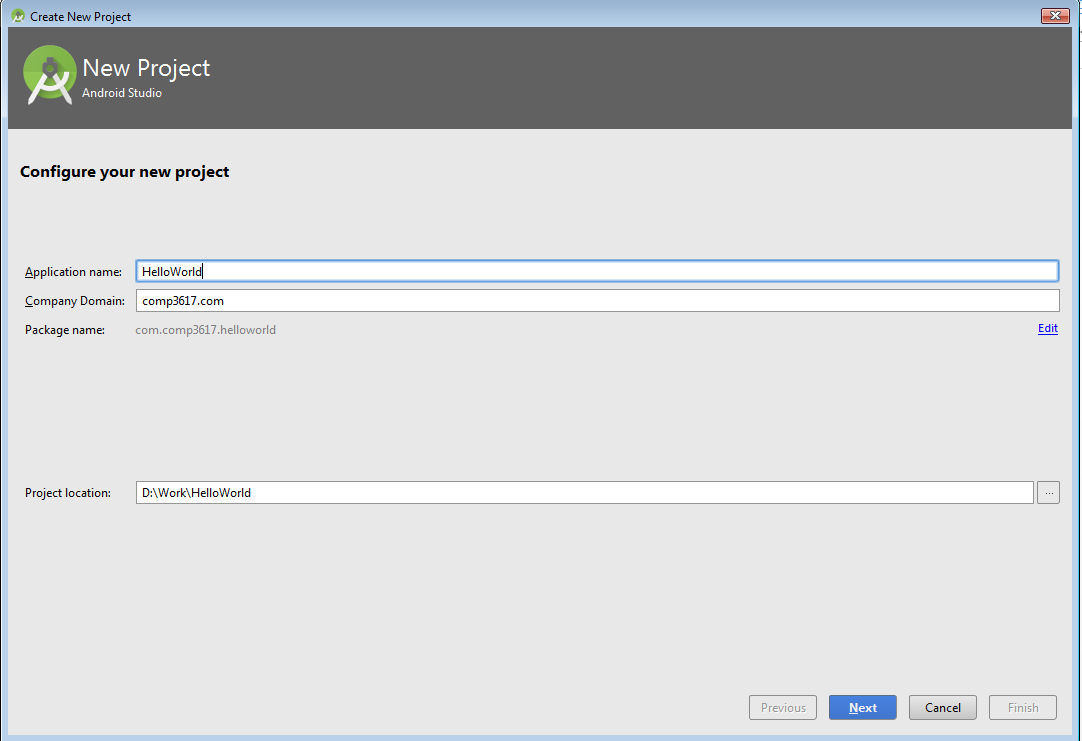
Click on “Install 8 Packages” button

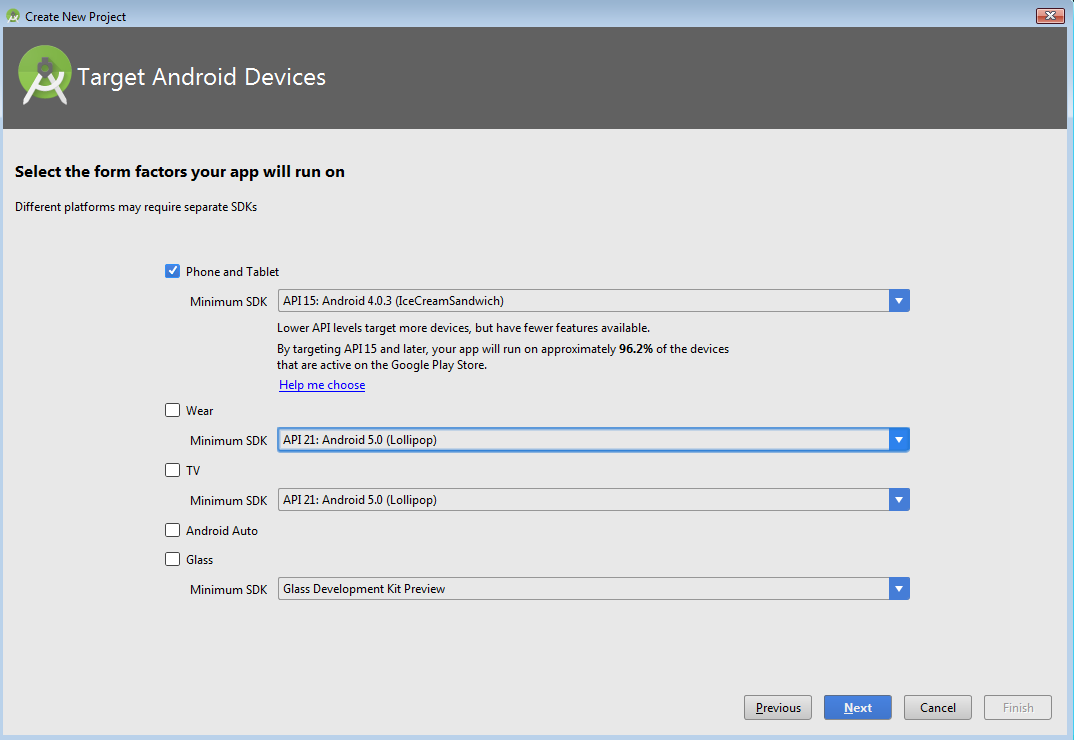


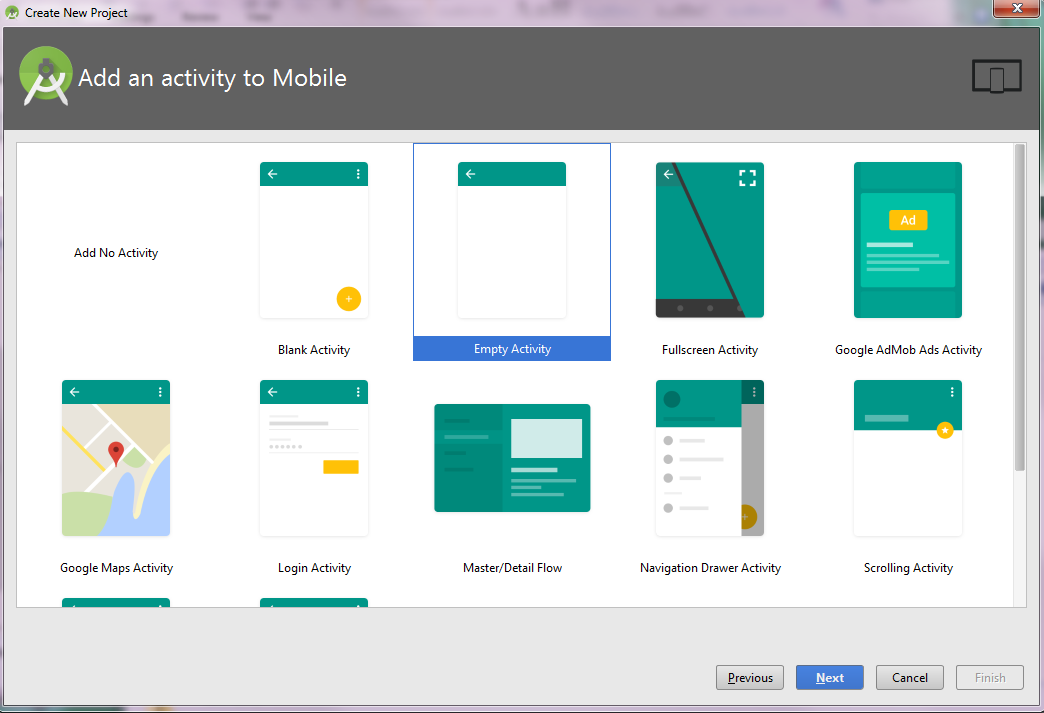


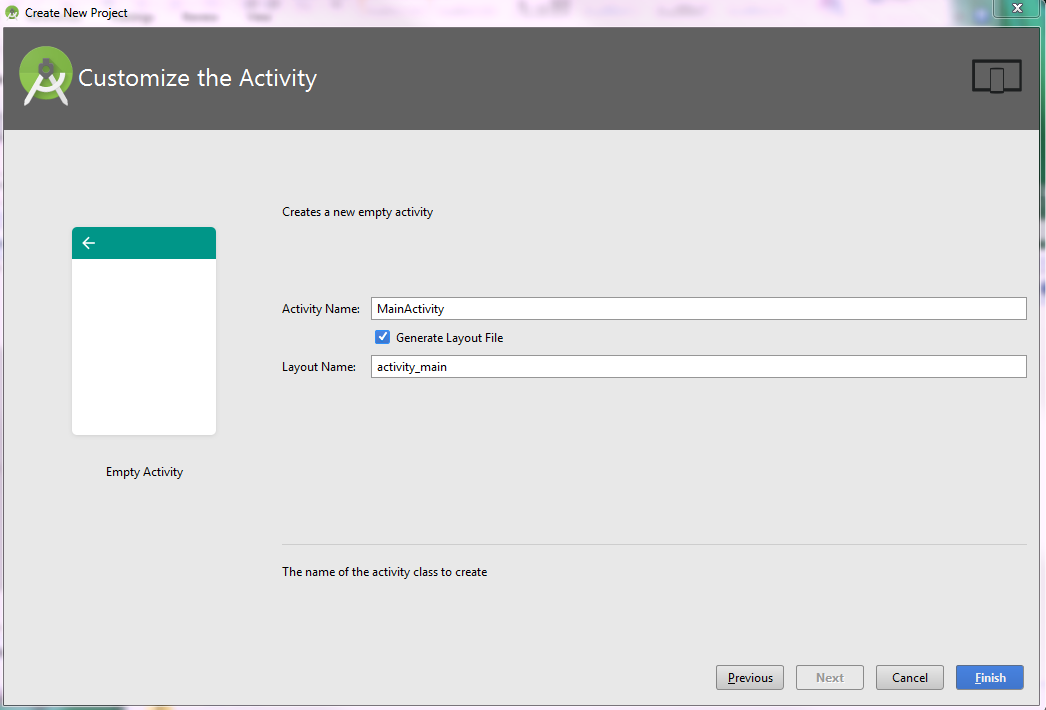
Takes a few minutes to download and install the software

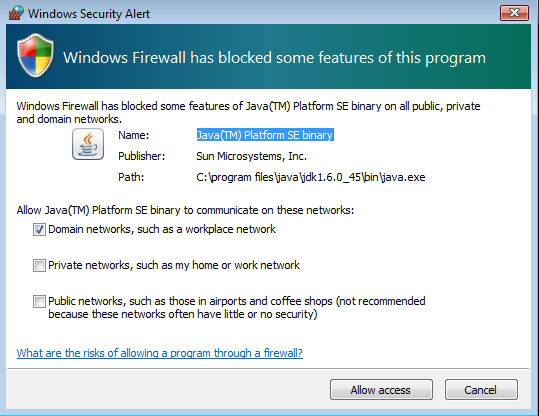
## Creating Android Virtual Devices (AVD)





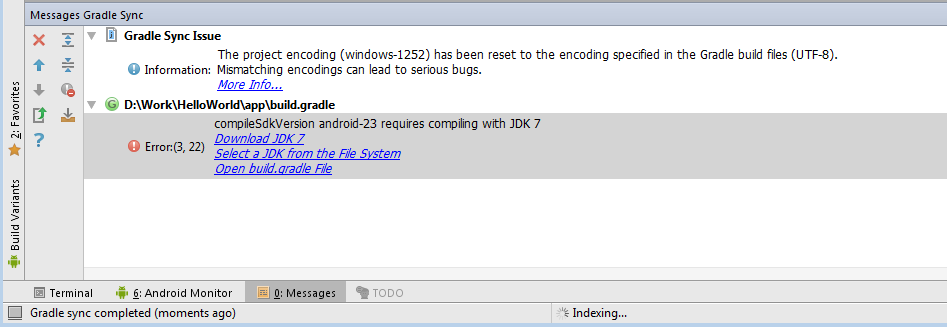


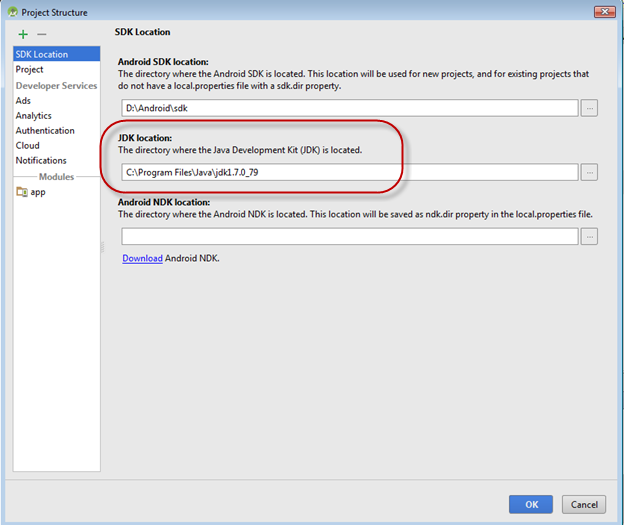




Allow access if the dialog shows up

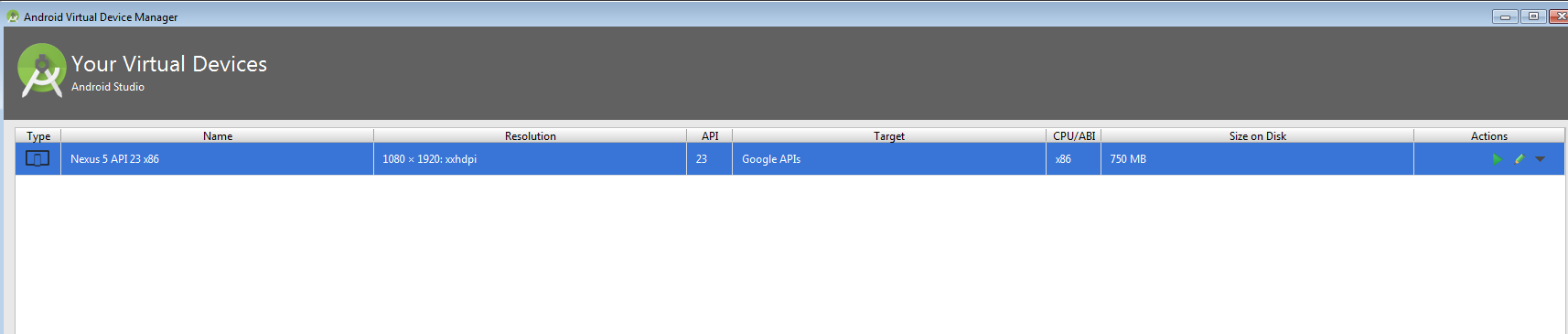
If you see the following image, Android Studio is not configured to use JDK 7. Click on the link “Select JDK from the file System”



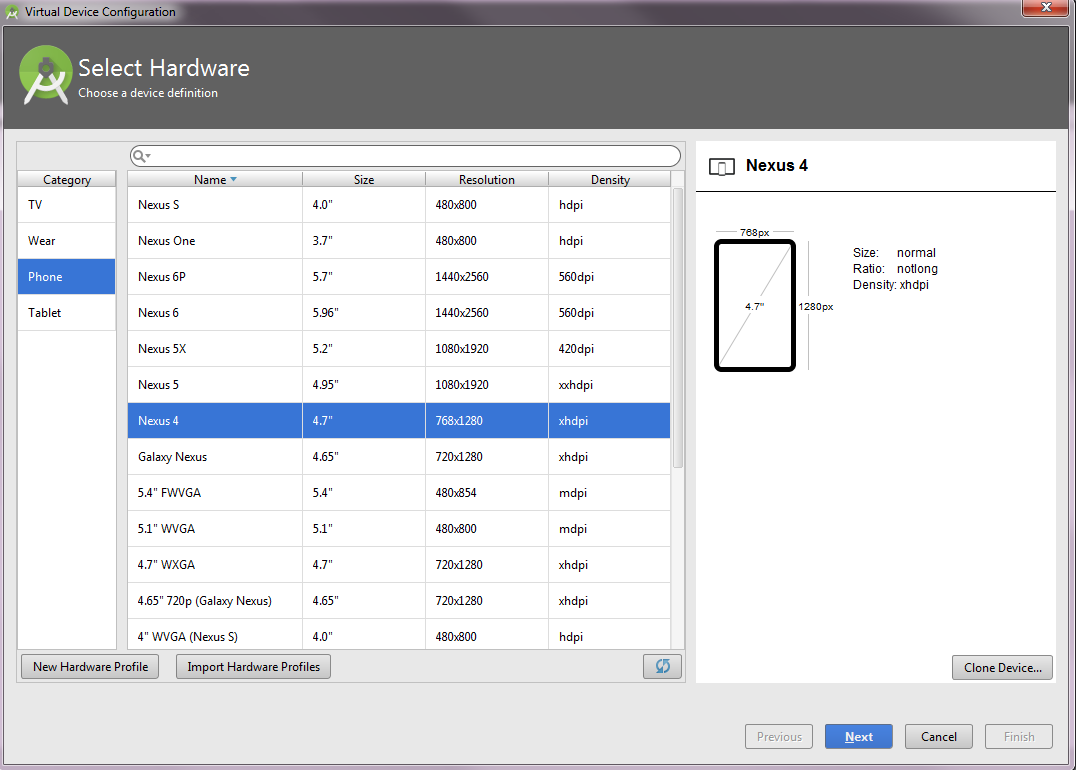


Provide the JDK location as shown above

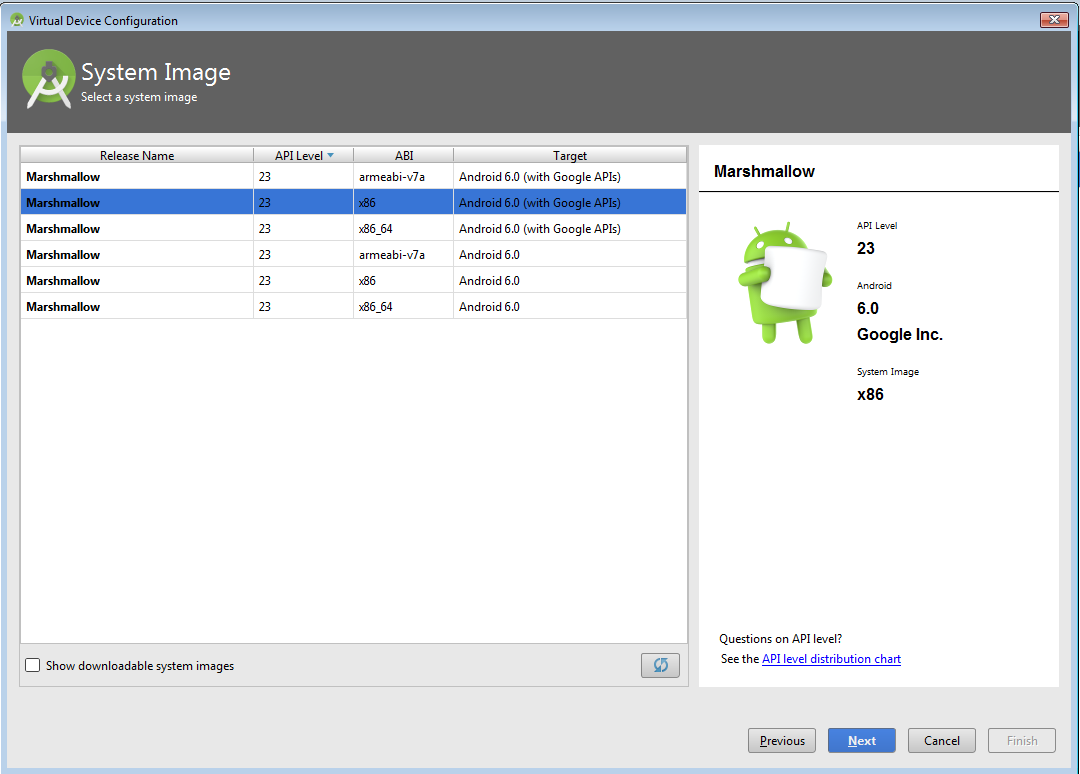
To launch the AVD go to Tools 🡪 Android 🡪 AVD Manager. You will notice there is a default one already created as part of the installation



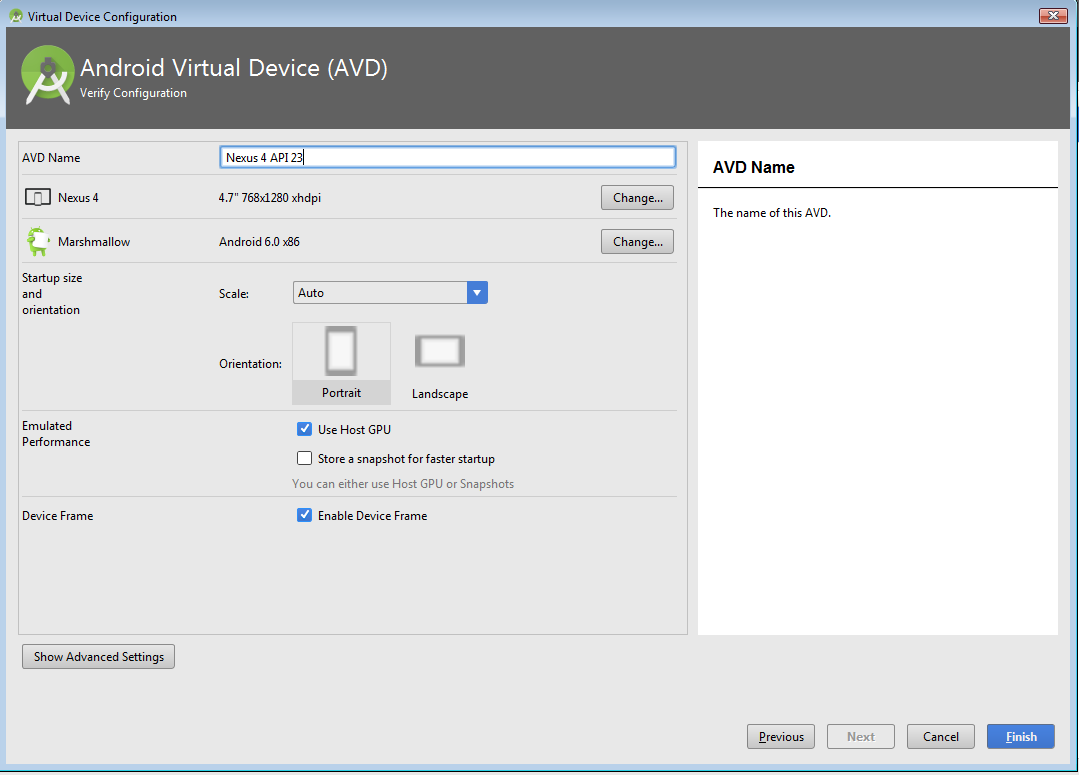
Click on the  to create a new AVD

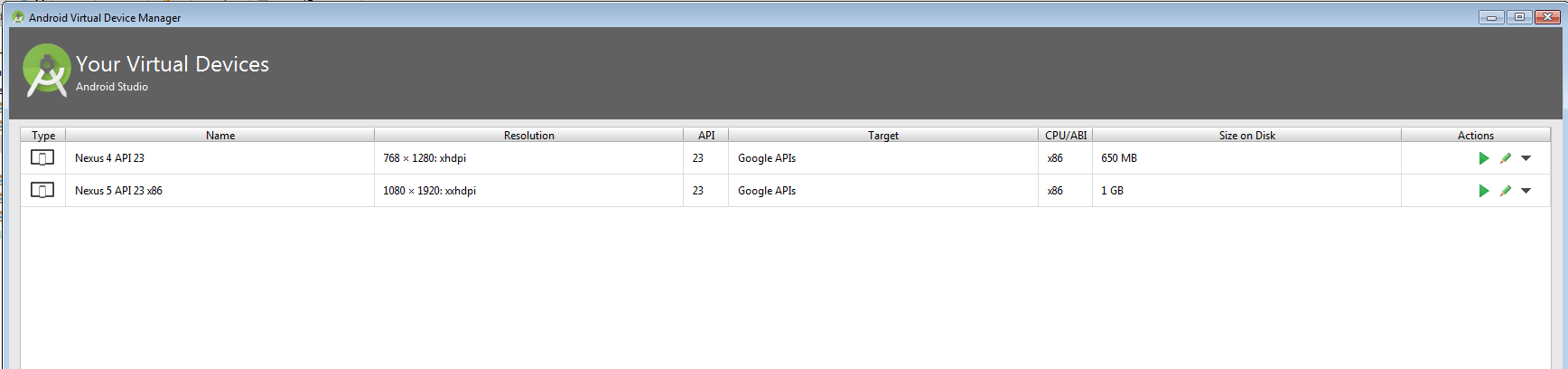


Select Nexus 4

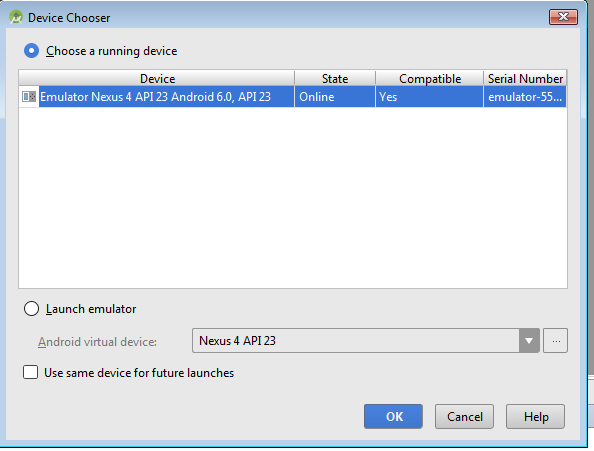


Choose Marshmallow and x86 with Google APIs image as shown above. We will pick the defaults for the next screen.

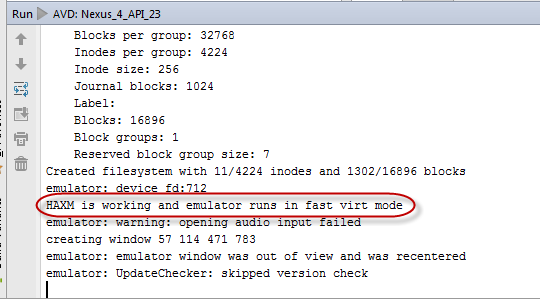




Click the Run button in your project () and it will prompt you to choose the AVD or device. Choose the AVD you just created



You should see the following information in Android Studio at the bottom of the screen





Here is the demo app

