

Android Application Development, COMP 10073

Mohawk College, Winter 2021

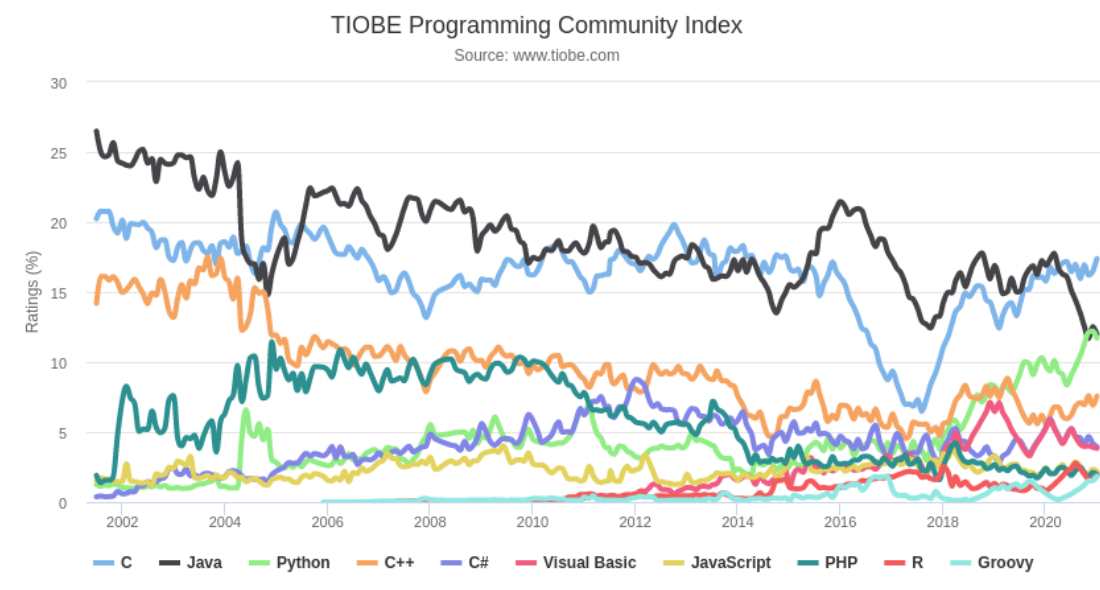
Introduction

Android is a mobile operating system based on a modified version of the Linux Kernel. Linus Torvalds is the author and principal developer of the Linux Kernel. Version 1.0 of Linux was released on 14th of March 1994. Torvalds has received numerous IEEE awards in recognition of his contributions to open source.



Many Android applications are written in Java, and for much of this course our focus will be learning the Java interface to the Android GUI. While Android applications can be written in a variety of languages, most are written in Java.

Java has been one of the top 3 most popular languages for the last 20 years.



For the last 10 years Java's popularity has been driven in part by Android.

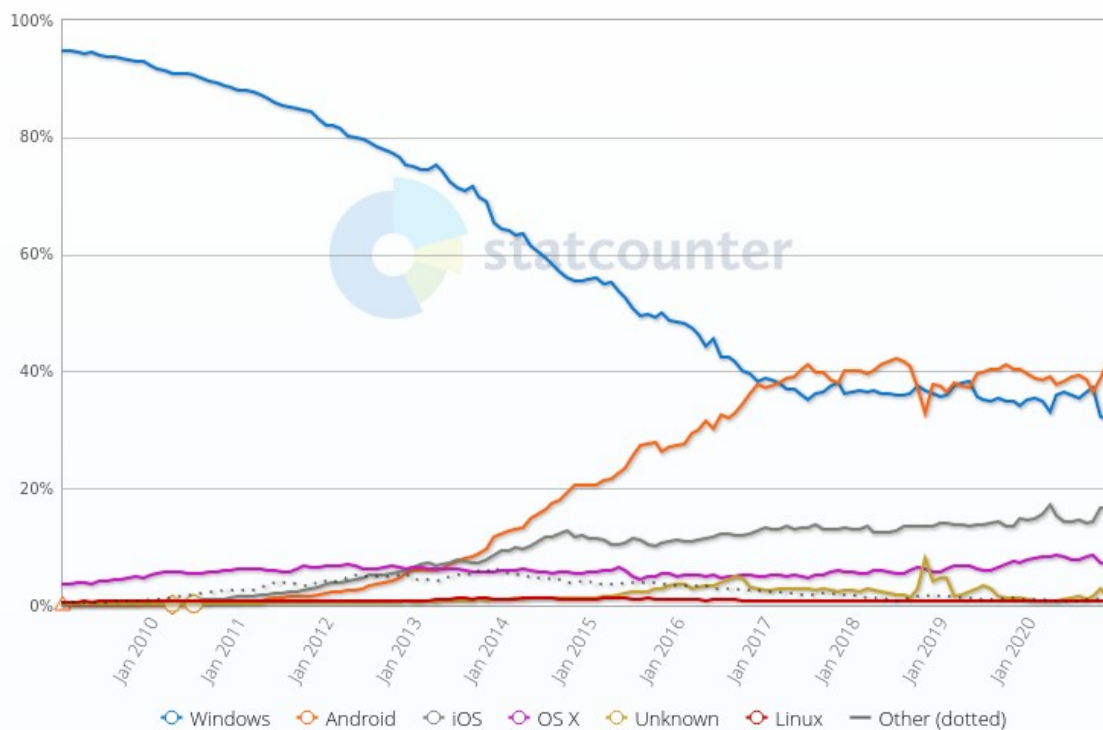
Today, Android is the most popular operating system in the world:



Since Android's initial release in 2008 it has steadily consumed Windows market share. Windows keeps its foothold partly through MS Office and its large legacy base of business users. Many average users no longer own a Windows device.

Operating System Market Share Worldwide

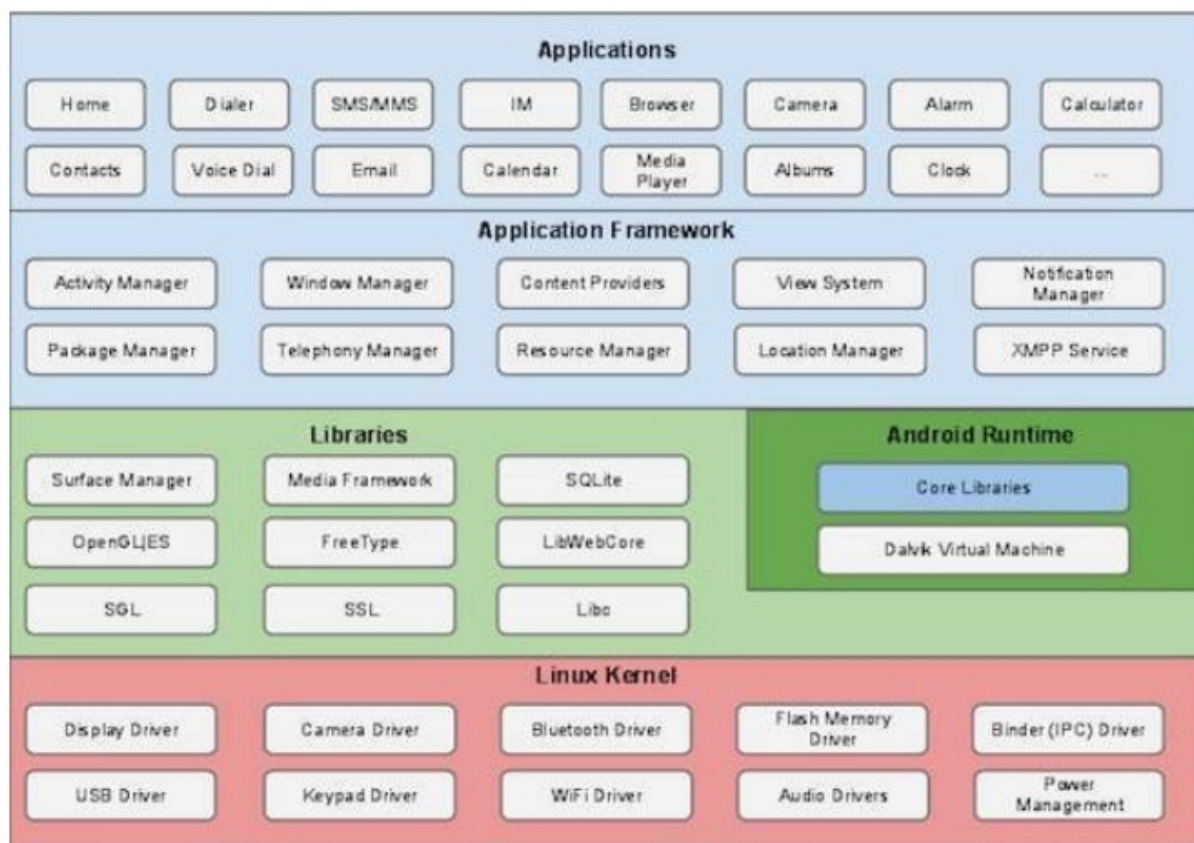
Jan 2009 - Dec 2020



While many people think of Linux as a server based operating system, Linux is also small, fast, efficient, adaptable and forms the basis of many embedded systems and small footprint devices.

Android and Linux work together on mobile devices, where the Linux OS defines the basic file system, input/output, and low level hardware interface for mobile devices and Android provides the user interface or service level access.

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



The application Framework separates the Android OS from what we would think of as the Linux OS. You can install SSH on an Android device and log into it and you will find that the file system looks very similar to a Linux Server.

In this course we will focus on developing simple programs at the application layer that make use of the Android Application Framework. We will discuss The Activity Manager, the View System, Notifications and other key components of the application framework and how to use these elements to build Android apps.

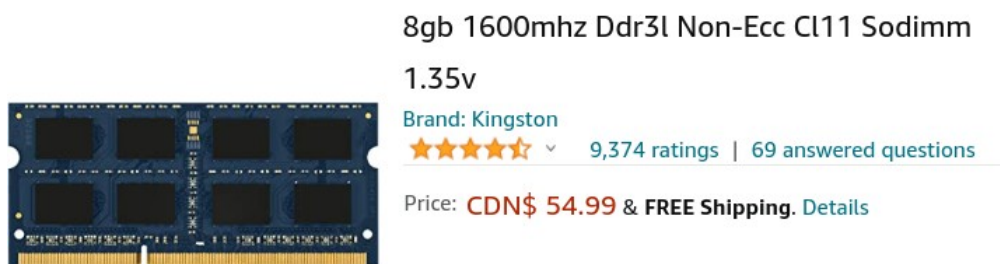
We will not dig into the lower layers. While Android is based on Linux, if you are familiar with the Linux file system, shell, or writing scripts for Linux this will not help you. Android is its own complex beast and learning the basics will take 14 weeks.

Android Studio

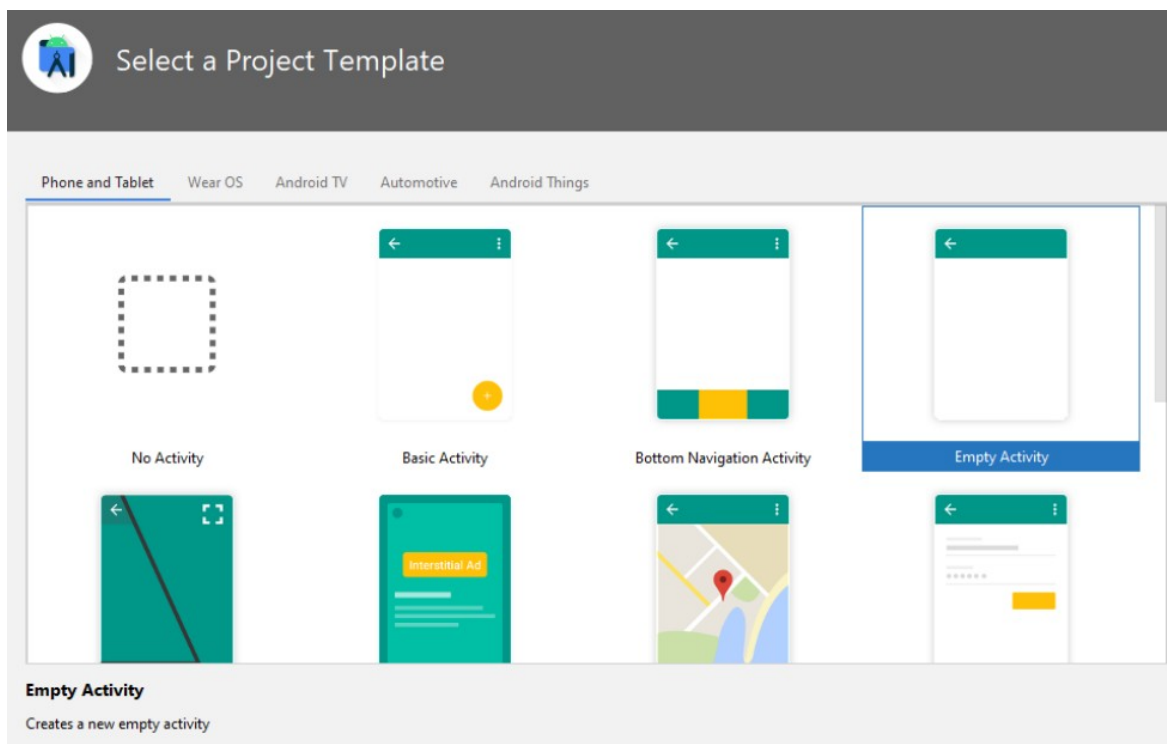
We will be using Android Studio to develop Java applications. Go to the Android Developer Web Site and download a copy for your computer. Cross platform support is very good. The Java compiler usually works out of the box.

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

Make sure that your computer has at least 8GB of RAM. If you are using an older laptop that has less memory than this Android Studio will not run properly on your machine. Check your user manual before you purchase memory as there are many varieties. The following module works in many laptops, and costs less than \$60.

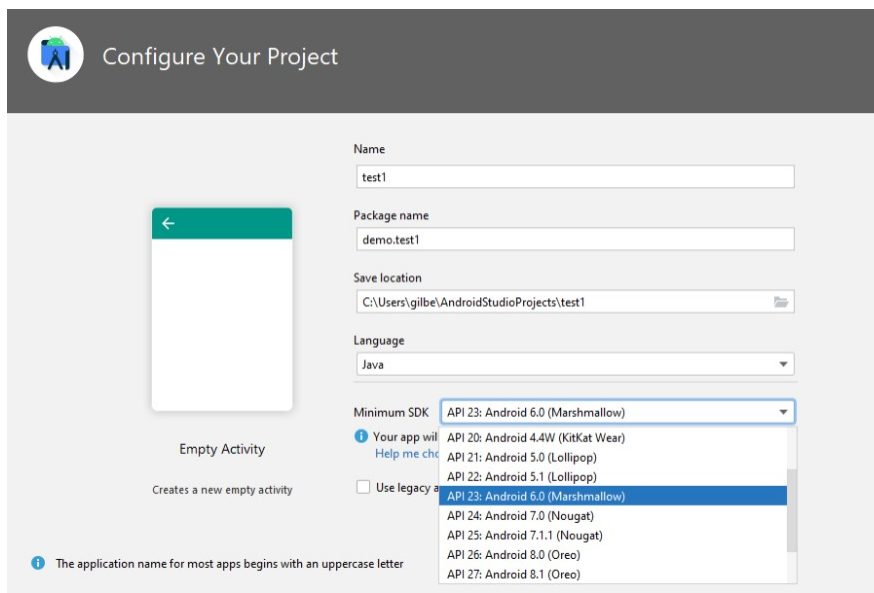


Create a new project using the **Empty Activity** template:



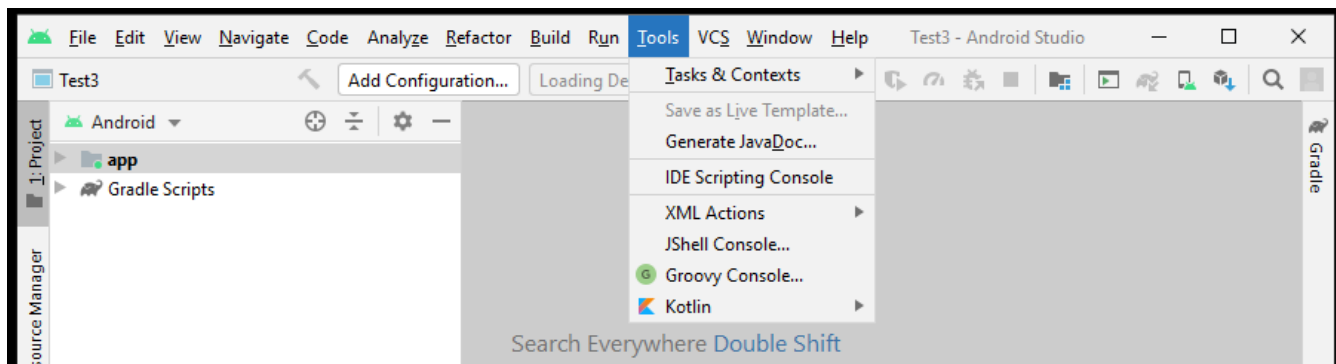
Configure your project, select a name, package name, choose “Java” for the language rather than “Kotlin”, and make sure you select:

API 23: Android 6.0 (Marshmallow)



All the examples in this course are based on API 23: Android 6.0. While they may work on other APIs Android is not fully backwards compatible. Many students like to select a version that they are familiar with, and this won't help you in this course.

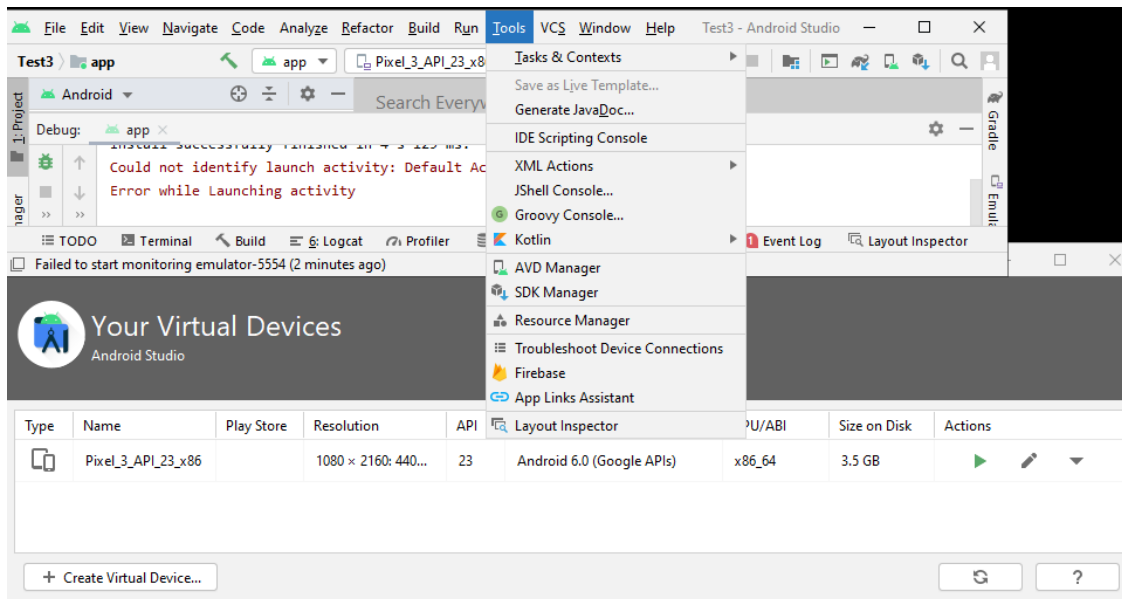
The tools menu will often be empty immediately after startup:



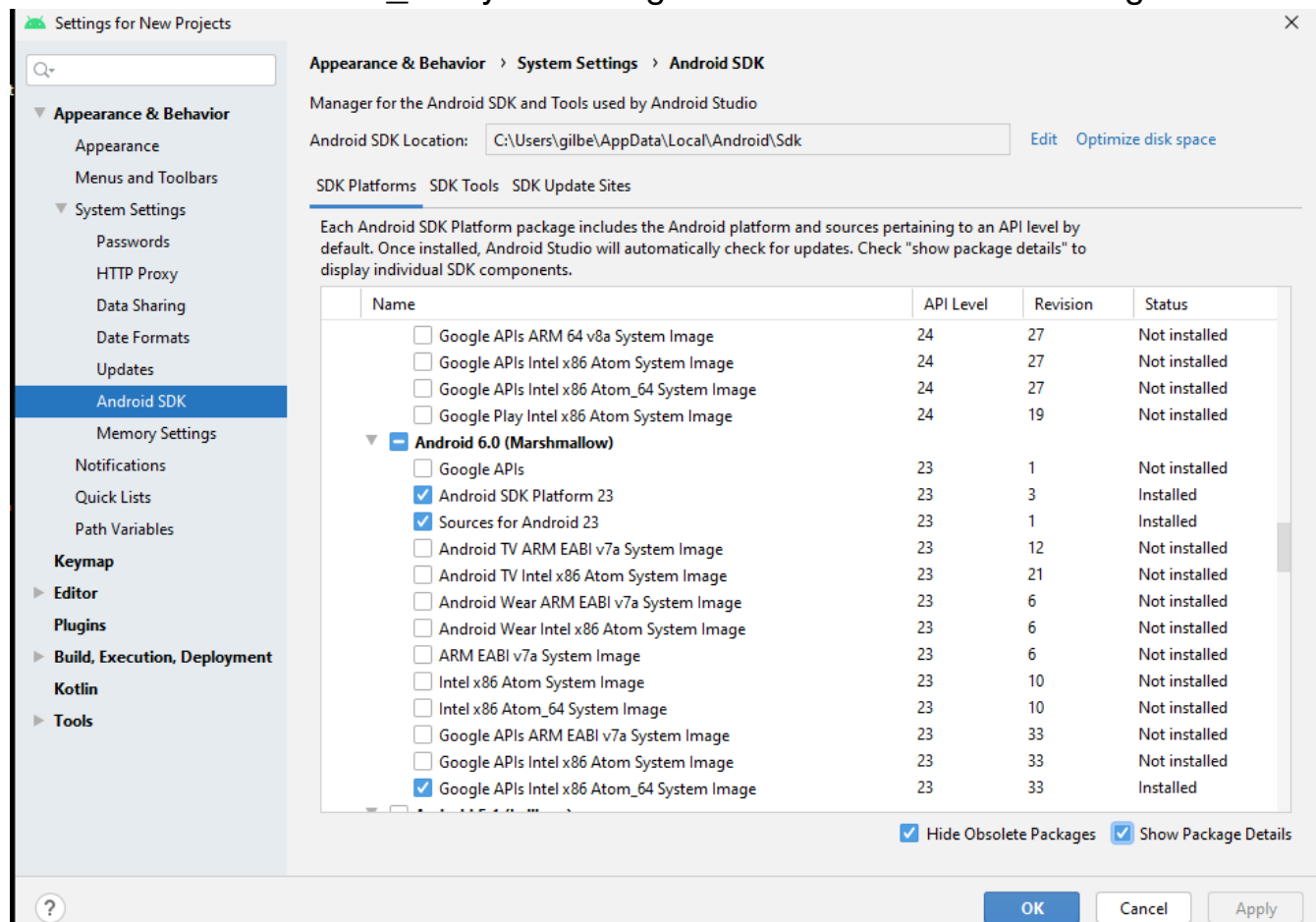
Usually you will need to wait while several background tasks run before Android Studio will show you all the menu items. If you check the messages at the bottom of the window you will see that Android Studio is indexing files, downloading packages, configuring Gradle, all in the background. It can take a while.

Android Studio will normally install a virtual Android device for the latest SDK. We need to ensure that we have a virtual device set up that matches SDK 23.

Start the AVD Manager and the SDK manager from the Tools menu once the system is “warmed” up.




Using the SDK manager, select “Show Package Details”. Ensure that you have Revision 33 of the Atom_64 System Image. Do not install the ARM image.



You must select a system image, again, go with **Marshmallow**:

Select a system image
Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
Nougat Download	24	x86	Android 7.0 (Google APIs)
Nougat Download	24	x86_64	Android 7.0 (Google APIs)
Nougat Download	24	x86_64	Android 7.0
Nougat Download	24	x86	Android 7.0
Marshmallow	23	x86_64	Android 6.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0
Marshmallow Download	23	x86_64	Android 6.0

Marshmallow

API Level **23**
Android **6.0**
Google Inc.
System Image **x86_64**
[Questions on API level?](#)
[See the API level distribution chart](#)

Which device you choose to emulate is not too important important. For consistency I'll suggest the Pixel 2. Some people have difficulty with the hardware graphics option, select software rendering if your emulator doesn't work.

Verify Configuration

AVD Name: Pixel 2 API 23

Pixel 2

5.0 1080x1920 xxhdpi

Change...

Marshmallow

Android 6.0 x86_64

Change...

Startup orientation

Portrait





Landscape

Emulated Performance
Graphics: Software - GLES 2.0



Graphics Rendering
Choose how the graphics should be rendered in the emulator.
Hardware
Use your computer's graphics card for faster rendering.
Software
Emulate the graphics in software, use this to work around issues with your computer's graphics card.

Once your device is setup and ready to go select cold boot from the drop down menu. You may need to cold boot your device periodically to handle errors.

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel_3_API_23_x86		1080 x 2160: 440dpi	23	Android 6.0 (Goo...	x86_64	3.5 GB	  

+ Create Virtual Device...

AndroidManifest.xml

java

Duplicate

Wipe Data

Cold Boot Now

Show on Disk

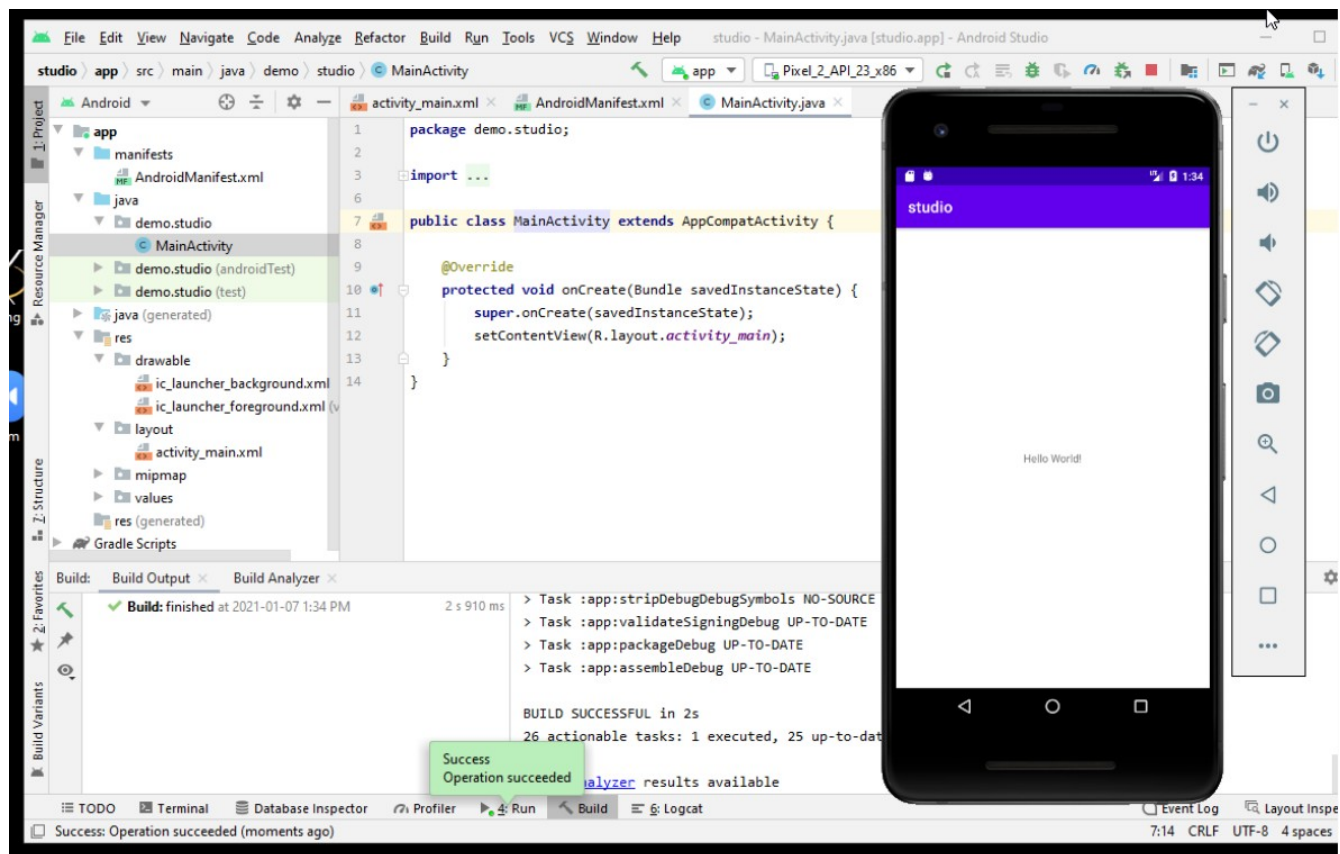
View Details

Delete

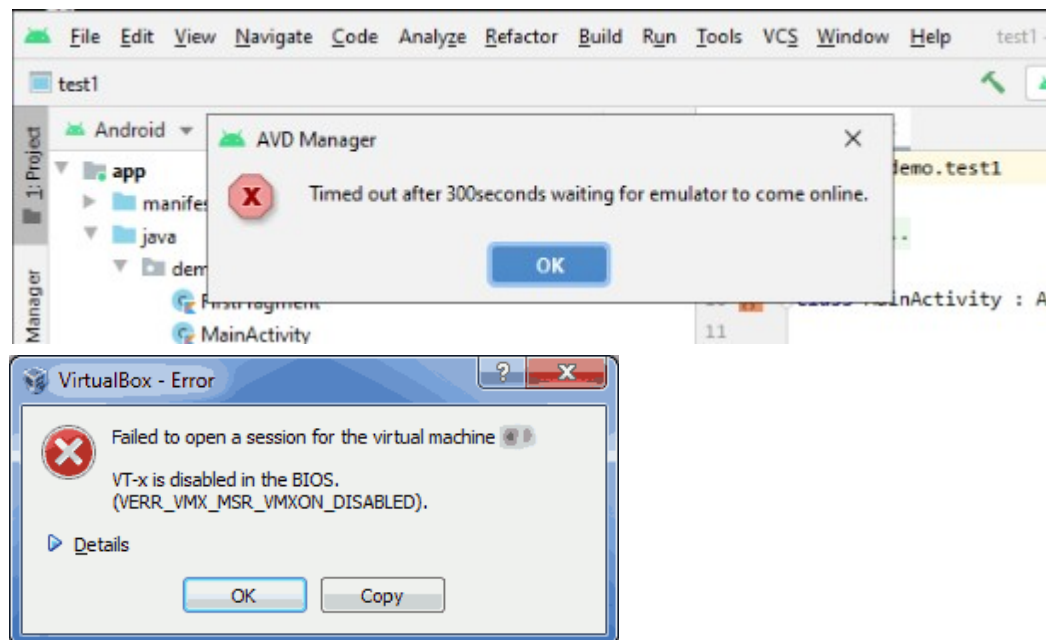
Stop

Normally the AVD will attempt a “warm boot” which is similar to waking the device from sleep. This saves time but will fail if the device crashes. You can wipe the data on the device or try a cold boot if you mess up your emulator.

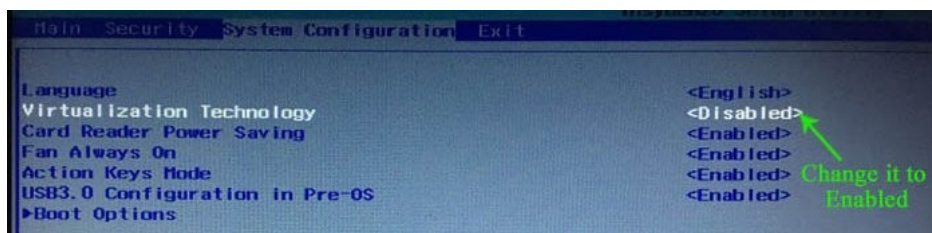
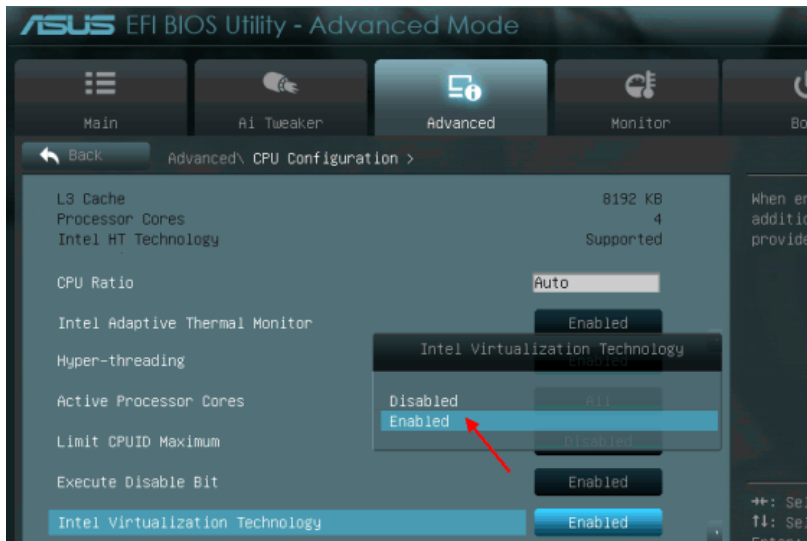
If you are lucky everything should work and you will have an emulated marshmallow 6 device to experiment with running a basic hello world app.



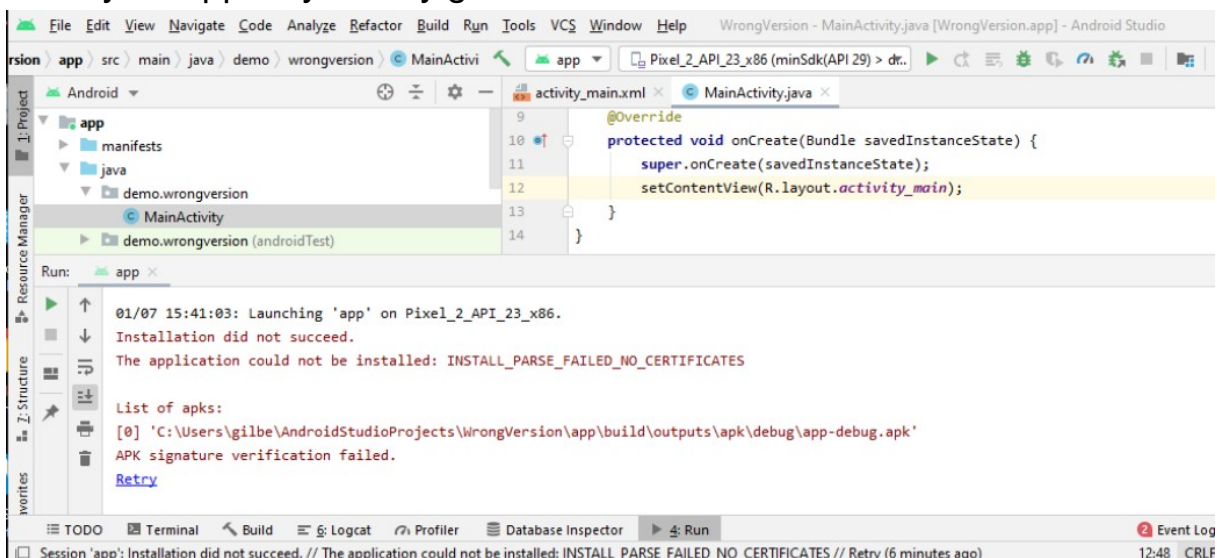
You may also get errors.



If you get the VT-x error, or an error related to virtual machines you need to check your BIOS settings. Reboot your computer and go through the BIOS configuration menus, verify that VT-x is properly enabled in your BIOS.

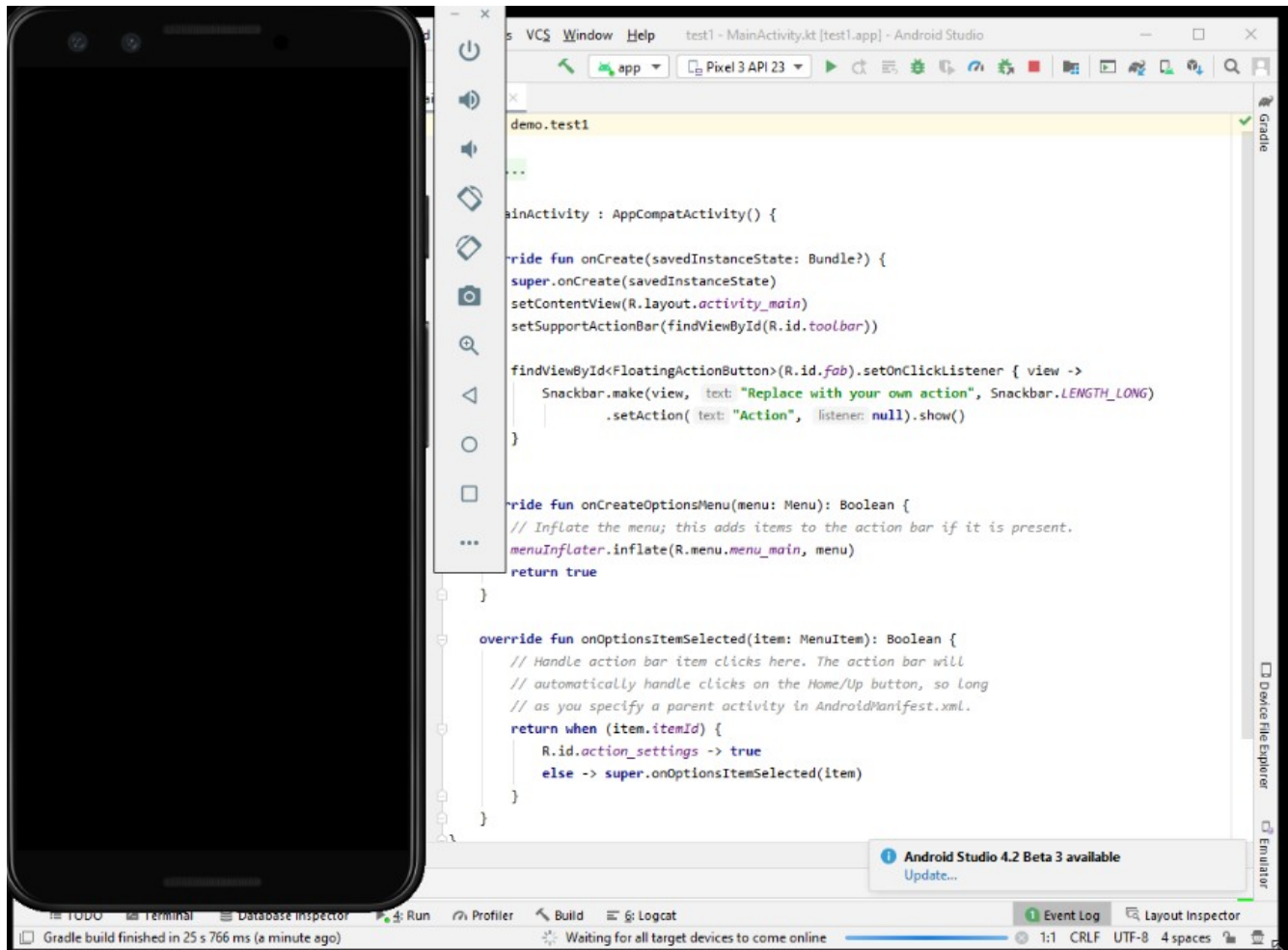


If you build your application with a higher SDK than the device you are trying to install your app on you may get an error like this:



Review your AVD settings and make sure the SDK versions are consistent.

In some cases the AVD will simply hang and not start. If your virtual device worked previously sometimes you can address this problem with a cold boot, or by wiping the data on the device. If this is your first try the problem may be the result of hardware incompatibility.



If you are having problems getting the AVD to work you may find it straightforward to install apps to a hardware device, however be warned that this should be considered a short term workaround.

I need to be able to compile your term tests, final project and final exam. I will use Android Studio 4.1.1 and a virtual Marshmallow device running Android 6.0 to test your implementation.

For any evaluation if your application cannot be tested on an emulated Android 6.0 device, i.e. it crashes, generates version errors, or refuses to start, you will not get a passing grade for that lab, test, or assignment.