V Skills

Skills for a secure future

Certified Android Apps Developer VS-1044

V-Skills Certifications

A Government of India

&

Government of NCT Delhi Initiative

V-Skills

Skills for a secure future

1. INTRODUCTION

Android is the prominent operating system for mobile devices specially the mobile phones and it commands a good number of devices in use.

1.1. Mobile Application Development

Mobile application development is the process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones. These applications can be pre-installed on phones during manufacturing, downloaded by customers from various mobile software distribution platforms, or delivered as web applications using server-side or client-side processing (e.g. JavaScript) to provide an "application-like" experience within a Web browser. Application software developers also have to consider a lengthy array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms. Mobile app development has been steadily growing, both in terms of revenues and jobs created. A 2013 analyst report estimates there are 529,000 direct App Economy jobs within the EU 28 members, 60% of which are mobile app developers.

As part of the development process, mobile user interface (UI) design is also an essential in the creation of mobile apps. Mobile UI considers constraints & contexts, screen, input and mobility as outlines for design. The user is often the focus of interaction with their device, and the interface entails components of both hardware and software. User input allows for the users to manipulate a system, and device's output allows the system to indicate the effects of the users' manipulation. Mobile UI design constraints include limited attention and form factors, such as a mobile device's screen size for a user's hand(s). Mobile UI contexts signal cues from user activity, such as location and scheduling that can be shown from user interactions within a mobile application. Overall, mobile UI design's goal is primarily for an understandable, user-friendly interface. The UI of mobile apps should: consider users' limited attention, minimize keystrokes, and be task-oriented with a minimum set of functions.

1.2. What is Android

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, regular PCs and other electronics.

Android is the most widely used mobile OS and, as of 2013, the highest selling OS overall. Android devices sell more than Windows, iOS, and Mac OS X devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013 the Google Play store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for

Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users, up from 538 million in June 2013.

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance—a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.

1.3. History of Android Platform

Android, Inc. was founded in Palo Alto, California in October 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile), and Chris White (headed design and interface development at WebTV) to develop, in Rubin's words, "smarter mobile devices that are more aware of its owner's location and preferences". The early intentions of the company were to develop an advanced operating system for digital cameras, when it was realized that the market for the devices was not large enough, and diverted their efforts to producing a smartphone operating system to rival those of Symbian and Windows Mobile. Despite the past accomplishments of the founders and early employees, Android Inc. operated secretly, revealing only that it was working on software for mobile phones. That same year, Rubin ran out of money. Steve Perlman, a close friend of Rubin, brought him $10,000 in cash in an envelope and refused a stake in the company.

Google acquired Android Inc. on August 17, 2005; key employees of Android Inc., including Rubin, Miner, and White, stayed at the company after the acquisition. Not much was known about Android Inc. at the time, but many assumed that Google was planning to enter the mobile phone market with this move. At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the promise of providing a flexible, upgradable system. Google had lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part.

Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. An earlier prototype codenamed "Sooner" had a closer resemblance to a BlackBerry phone, with no touchscreen, and a physical, QWERTY keyboard, but was later re-engineered to support a touchscreen, to compete with other announced devices such as the 2006 LG Prada and 2007 Apple iPhone. In September 2007, InformationWeek covered an Evalueserve study reporting that Google had filed several patent applications in the area of mobile telephony.

On November 5, 2007, the Open Handset Alliance, a consortium of technology companies including Google, device manufacturers such as HTC, Sony and Samsung, wireless carriers such as Sprint Nextel and T-Mobile, and chipset makers such as Qualcomm and Texas Instruments, unveiled itself, with a goal to develop open standards for mobile devices. That day, Android was unveiled as its first product, a mobile device platform built on the Linux kernel version 2.6.25. The first commercially available smartphone running Android was the HTC Dream, released on October 22, 2008.

In 2010, Google launched its Nexus series of devices – a line of smartphones and tablets running the Android operating system, and built by manufacturing partners. HTC collaborated with Google to release the first Nexus smartphone, the Nexus One. Google has since updated the series with newer devices, such as the Nexus 5 phone (made by LG) and the Nexus 7 tablet (made by Asus). Google releases the Nexus phones and tablets to act as their flagship Android devices, demonstrating Android's latest software and hardware features. On March 13, 2013 Larry Page announced in a blog post that Andy Rubin had moved from the Android division to take on new projects at Google. He was replaced by Sundar Pichai, who also continues his role as the head of Google's Chrome division, which develops Chrome OS.

Since 2008, Android has seen numerous updates which have incrementally improved the operating system, adding new features and fixing bugs in previous releases. Each major release is named in alphabetical order after a dessert or sugary treat; for example, version 1.5 Cupcake was followed by 1.6 Donut. The latest released version, 4.4.4 KitKat, appeared as a security-only update; it was released on June 19, 2014, shortly after the release of 4.4.3. As of October 2014, newest version of the Android operating system, Android 5.0 "Lollipop", is available only as a developer preview.

From 2010 to 2013, Hugo Barra served as product spokesperson for the Android team, representing Android at both press conferences and Google I/O, Google's annual developer-focused conference. Barra's product involvement included the entire Android ecosystem of software and hardware, including Honeycomb, Ice Cream Sandwich, Jelly Bean and KitKat operating system launches, the Nexus 4 and Nexus 5 smartphones, the Nexus 7 and Nexus 10 tablets, and other related products such as Google Now[53] and Google Voice Search, Google's speech recognition product comparable to Apple's Siri. In 2013 Barra left the Android team for Chinese smartphone maker Xiaomi.

Version History

The version history of the Android mobile operating system began with the release of the Android beta in November 2007. The first commercial version, Android 1.0, was released in September 2008. Android is under ongoing development by Google and the Open Handset Alliance (OHA), and has seen a number of updates to its base operating system since its initial release.

Since April 2009, Android versions have been developed under a confectionery-themed code name and released in alphabetical order; the exceptions are versions 1.0 and 1.1 as they were not released under specific code names:

- ✓ Alpha (1.0)
- ✓ Beta (1.1)

- ✓ Cupcake (1.5)
- ✓ Donut (1.6)
- ✓ Eclair (2.0–2.1)
- ✓ Froyo (2.2–2.2.3)
- ✓ Gingerbread (2.3–2.3.7)
- ✓ Honeycomb (3.0–3.2.6)
- ✓ Ice Cream Sandwich (4.0–4.0.4)
- ✓ Jelly Bean (4.1–4.3.1)
- ✓ KitKat (4.4–4.4.4)
- ✓ Lollipop (5.0)

On September 3, 2013, Google announced that one billion activated devices now use the Android OS worldwide. The most recent major Android update was Lollipop 5.0, which was released on November 3, 2014 along with the Nexus 6 smartphone, Nexus 9 Tablet, and Nexus Player set-top box

1.4. Android Features

General

- ✓ Messaging - SMS and MMS are available forms of messaging, including threaded text messaging and Android Cloud To Device Messaging (C2DM) and now enhanced version of C2DM, Android Google Cloud Messaging (GCM) is also a part of Android Push Messaging service.
- ✓ Web browser- The web browser available in Android is based on the open-source Blink (previously WebKit) layout engine, coupled with Chrome's V8 JavaScript engine. The browser scores 100/100 on the Acid3 test on Android 4.0.
- ✓ Voice based features - Google search through voice has been available since initial release. Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards. As of Android 4.1, Google has expanded Vce Actions with ability to talk back and read answers from Google's Knowledge Graph when queried with specific commands. The ability to control hardware has not yet been implemented.
- ✓ Multi-touch - Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. The feature was originally disabled at the kernel level (possibly to avoid infringing Apple's patents on touch-screen technology at the time). Google has since released an update for the Nexus One and the Motorola Droid which enables multi-touch natively.
- ✓ Multitasking - Multitasking of applications, with unique handling of memory allocation, is available.
- ✓ Screen capture - Android supports capturing a screenshot by pressing the power and volume-down buttons at the same time. Prior to Android 4.0, the only methods of capturing a screenshot were through manufacturer and third-party customizations or otherwise by using a PC connection (DDMS developer's tool). These alternative methods are still available with the latest Android.
- ✓ Video calling - Android does not support native video calling, but some handsets have a customized version of the operating system that supports it, either via the UMTS network (like the Samsung Galaxy S) or over IP. Video calling through Google Talk is available in Android 2.3.4 and later. Gingerbread allows Nexus S to place Internet calls with a SIP account. This

allows for enhanced VoIP dialing to other SIP accounts and even phone numbers. Skype 2.1 offers video calling in Android 2.3, including front camera support. Users with the Google+ Android app can video chat with other google+ users through hangouts.

- ✓ Multiple language support - Android supports multiple languages.
- ✓ Accessibility - Built in text to speech is provided by Talk back for people with low or no vision. Enhancements for people with hearing difficulties are available as are other aids.

Connectivity

- ✓ Connectivity - Android supports connectivity technologies including GSM/EDGE, Wi-Fi, Bluetooth, LTE, CDMA, EV-DO, UMTS, NFC, IDEN and WiMAX.
- ✓ Bluetooth - Supports voice dialing and sending contacts between phones, sending files (OPP), accessing the phone book (PBAP), A2DP and AVRCP. Keyboard, mouse and joystick (HID) support is available in Android 3.1+, and in earlier versions through manufacturer customizations and third-party applications.
- ✓ Tethering - Android supports tethering, which allows a phone to be used as a wireless/wired Wi-Fi hotspot. Before Android 2.2 this was supported by third-party applications or manufacturer customizations.

Media

- ✓ Streaming media support - RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP) and HTTP Dynamic Streaming are supported by the Flash plugin. Apple HTTP Live Streaming is supported by RealPlayer for Android, and by the operating system since Android 3.0 (Honeycomb).
- ✓ Media support - Android supports the following audio/video/still media formats: WebM, H.263, H.264, AAC, HE-AAC (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), MP3, MIDI, Ogg Vorbis, FLAC, WAV, JPEG, PNG, GIF, BMP, WebP.
- ✓ External storage - Most Android devices include microSD slot and can read microSD cards formatted with FAT32, Ext3 or Ext4 file system. To allow use of high-capacity storage media such as USB flash drives and USB HDDs, many Android tablets also include USB 'A' receptacle. Storage formatted with FAT32 is handled by Linux Kernel VFAT driver, while 3rd party solutions are required to handle other popular file systems such as NTFS, HFS Plus and exFAT.

Hardware Support

Android devices can include still/video cameras, touchscreens, GPS, accelerometers, gyroscopes, barometers, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.

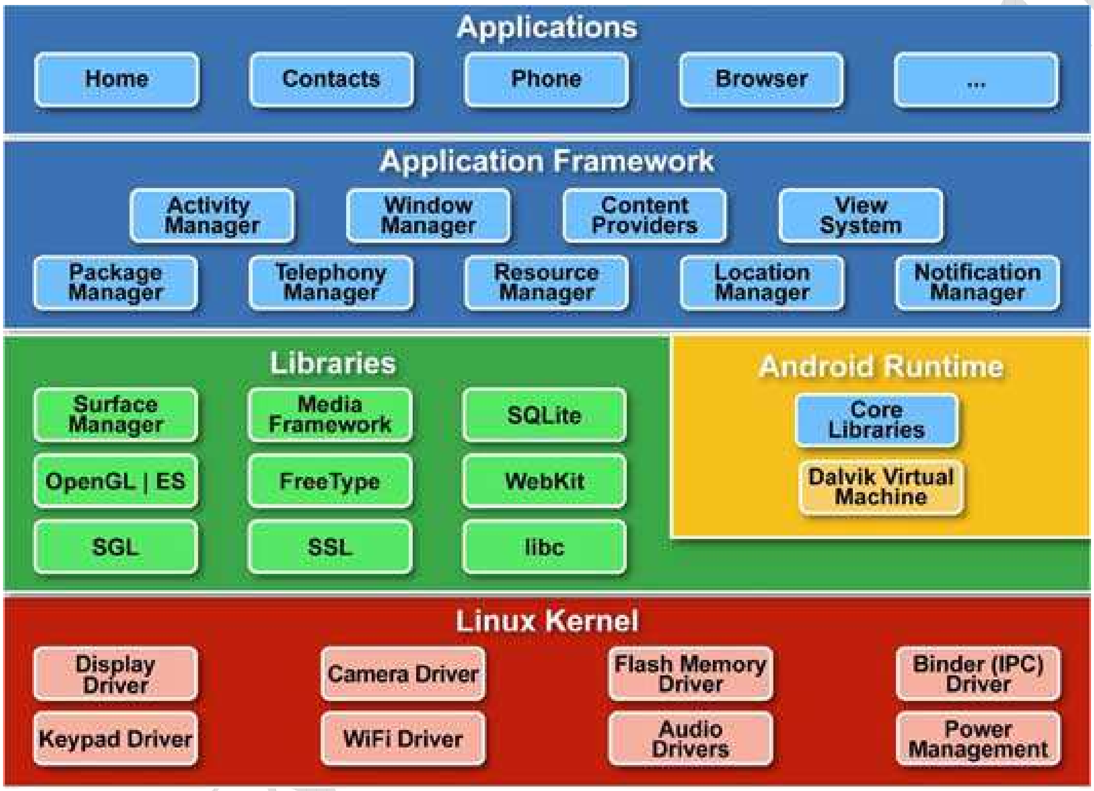
Other

- ✓ Java support - While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executables and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

- ✓ Handset layouts - The platform works for various screen sizes from smartphone sizes and to tablet size, and can potentially connect to an external screen, e.g. through HDMI, or wirelessly with Miracast. Portrait and landscape orientations are supported and usually switching between by turning. A 2D graphics library, 3D graphics library based on OpenGL ES 2.0 specifications is used.
- ✓ Storage - SQLite, a lightweight relational database, is used for data storage purposes.

1.5. Android Architecture

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.



Linux Kernel

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

1.6. Pre-requisites for Development

Android applications are developed using Java and can be ported rather easily to the new platform. Android is made up of several necessary and dependent parts including the following:

- ✓ A hardware reference design that describes the capabilities required of a mobile device in order to support the software stack
- ✓ A Linux operating system kernel that provides the low-level interface with the hardware, memory management, and process control, all optimized for mobile devices
- ✓ Open source libraries for application development including SQLite, WebKit, OpenGL, and a media manager
- ✓ A run time used to execute and host Android applications, including the Dalvik virtual machine and the core libraries that provide Android specific functionality. The run time is designed to be small and efficient for use on mobile devices.
- ✓ An application framework that agnostically exposes system services to the application layer, including the window manager, content providers, location manager, telephony, and peer-to-peer services
- ✓ A user interface framework used to host and launch applications
- ✓ Preinstalled applications shipped as part of the stack
- ✓ A software development kit used to create applications, including the tools, plug-ins, and documentation

The Android SDK includes all the tools and APIs needed to write Android mobile applications. The biggest challenge with Android, as with any new development toolkit, is learning the features and limitations of its APIs. As it is based on Java programming language development hence, the techniques, syntax, and grammar in it will translate directly into Android. Even experience with any other object-oriented languages (such as C#), will help in same. The power of Android comes from its APIs, not from Java. To get started, download and install the following

- ✓ The Android SDK
- ✓ Java Development Kit (JDK) 5 or 6

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit, but other development tools are available.

Official Development Tools

Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows XP or later. For the moment one can also develop Android software on Android itself by using the AIDE - Android IDE - Java, C++ app and the Java editor app. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box, and NetBeans IDE also supports Android development via a plugin. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dex files (compiled byte code files called Dalvik executables), resource files, etc.

Android Debug Bridge

The Android Debug Bridge (ADB) is a toolkit included in the Android SDK package. It consists of both client and server-side programs that communicate with one another. The ADB is typically accessed through the command-line interface, although numerous graphical user interfaces exist to control ADB.

The format for issuing commands through the ADB is typically:

```
adb [-d|-e|-s <serialNumber>] <command>
```

In a security issue reported in March 2011, ADB was targeted as a vector to attempt to install a rootkit on connected phones using a "resource exhaustion attack"

Fastboot

Fastboot is a diagnostic protocol included with the SDK package used primarily to modify the flash filesystem via a USB connection from host computer. It requires that the device be started in a boot loader or Second Program Loader mode in which only the most basic hardware initialization is performed. After enabling the protocol on the device itself, it will accept a specific set of commands sent to it via USB using a command line. Some of the most commonly used fastboot commands include:

- ✓ flash – rewrites a partition with a binary image stored on the host computer
- ✓ erase – erases a specific partition
- ✓ reboot – reboots the device into either the main operating system, the system recovery partition or back into its bootloader
- ✓ devices – displays a list of all devices (with the serial number) connected to the host computer
- ✓ format – formats a specific partition; the file system of the partition must be recognized by the device

Native Development Kit

Libraries written in C, C++ and other languages can be compiled to ARM, MIPS or x86 native code and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the System.loadLibrary call, which is part of the standard Android Java classes.

Complete applications can be compiled and installed using traditional development tools. However, according to the Android documentation, NDK should not be used solely for developing applications only because the developer prefers to program in C/C++, as using NDK increases complexity while most applications would not benefit from using it.

The ADB debugger gives a root shell under the Android Emulator which allows ARM, MIPS or x86 native code to be uploaded and executed. Native code can be compiled using GCC or the Intel C++ Compiler on a standard PC. Running native code is complicated by Android's use of a non-standard C library (libc, known as Bionic). The graphics library that Android uses to arbitrate and control access to this device is called the Skia Graphics Library (SGL), and it has been released under an open source licence. Skia has backends for both Win32 and Unix, allowing the development of cross-platform applications, and it is the graphics engine underlying the Google Chrome web browser.

Unlike Java application development based on the Eclipse IDE, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.
Android Open Accessory Development Kit

The Android 3.1 platform (also backported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode.

Third-party Development Tools

Android APIMiner

Android APIMiner is a platform that automatically instruments the Javadoc documentation of the Android API with examples of usage, extracted from real open-source Android applications. To improve the quality of the extracted examples, APIMiner relies on an intra-procedural static slicing algorithm.

AndroWish

AndroWish is an on-device Tcl/Tk interpreter which provides multi-windows GUI and give access to Bluetooth, USB, network sockets, text-to-speech, speech recognition, sensor, location (GPS or network information), activity manager, SQLite and sound. The GUI layer relies on SDL2. AndroWish is provided with 337 Tcl packages.

App Inventor for Android

On July 12, 2010, Google announced the availability of App Inventor for Android, a Web-based visual development environment for novice programmers, based on MIT's Open Blocks Java library and providing access to Android devices' GPS, accelerometer and orientation data, phone functions, text messaging, speech-to-text conversion, contact data, persistent storage, and Web services, initially including Amazon and Twitter. "We could only have done this because Android’s architecture is so open," said the project director, MIT's Hal Abelson. Under development for over a year, the block-editing tool has been taught to non-majors in computer science at Harvard, MIT, Wellesley, Trinity College (Hartford,) and the University of San Francisco, where Professor David Wolber developed an introductory computer science course and tutorial book for non-computer science students based on App Inventor for Android.

In the second half of 2011, Google released the source code, terminated its Web service, and provided funding for the creation of The MIT Center for Mobile Learning, led by the App Inventor creator Hal Abelson and fellow MIT professors Eric Klopfer and Mitchel Resnick. Latest version created as the result of Google–MIT collaboration was released in February 2012, while the first version created solely by MIT was launched in March 2012 and upgraded to App Inventor 2 in December 2013. As of 2014, App inventor is now maintained by MIT.

Basic4android

Basic4android is a commercial product similar to Simple. It is inspired by Microsoft Visual Basic 6 and Microsoft Visual Studio. It makes android programming much simpler for regular Visual Basic programmers who find coding in Java difficult. Basic4android is very active, and there is a strong online community of Basic4android developers.

Kivy

Kivy is an open source Python library for developing multitouch application software with a natural user interface (NUI) for a wide selection of devices. Kivy provides the possibility of maintaining a single application for numerous operating systems ("code once, run everywhere"). Kivy has a custom-built deployment tool for deploying mobile applications called Buildozer, which is available only for Linux. Buildozer is currently alpha software, but is far less cumbersome than older Kivy deployment methods. Applications programmed with Kivy can be submitted to any Android mobile application distribution platform.

Processing

The Processing environment, which also uses the Java language, has supported an Android mode since version 1.5; integration with device camera and sensors is possible using the Ketai library.

Saphir

Saphir is a fork of the open source Rebol 3 (R3) interpreter. All R3 functionality, including GUI, graphics, network access, file access, parse, and other core features are portable across Android, Windows, Mac, and Linux platforms, without any changes to source code. Saphir makes use of concise dialect (DSL) code patterns to build graphic user interfaces and to perform common computational operations, with remarkably short and readable code syntax. The trivial size of the interpreter (.5 - 1.5 megabytes) complements Saphir's simple utilitarian design.

SDL

The SDL library offers also a development possibility beside Java, allowing the development with C and the simple porting of existing SDL and native C applications. By injection of a small Java shim and JNI the usage of native SDL code is possible, allowing Android ports like e.g. the Jagged Alliance 2 video game.

The Simple project

The goal of Simple is to bring an easy-to-learn-and-use language to the Android platform. Simple is a BASIC dialect for developing Android applications. It targets professional and non-professional programmers alike in that it allows programmers to quickly write Android applications that use the Android runtime components.

Similar to Microsoft Visual Basic 6, Simple programs are form definitions (which contain components) and code (which contains the program logic). The interaction between the components and the program logic happens through events triggered by the components. The program logic consists of event handlers which contain code reacting to the events.

The Simple project is not very active, the last source code update being in August 2009.

WinDev Mobile

WinDev Mobile is a proprietary IDE created by PC SOFT and is used to create graphical user interface (GUI) applications for smartphones and tablets (including Android devices). It uses the WLanguage as its programming language, which is available in English, French and Chinese.

1.7. Environment Setup

Before development, the development environment need to be set up which needs

- ✓ Download the Android SDK.
- ✓ Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
- ✓ Download the latest SDK tools and platforms using the SDK Manager.

Install the most recent versions of the ADT plugin and the Android SDK. Android application development can be done on the following operating systems

- ✓ Microsoft Windows XP or later version.
- ✓ Mac OS X 10.5.8 or later version with Intel chip.
- ✓ Linux including GNU C Library 2.7 or later.

All the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software needed for Android application programming.

- ✓ Java JDK5 or JDK6
- ✓ Android SDK
- ✓ Eclipse IDE for Java Developers (optional)
- ✓ Android Development Tools (ADT) Eclipse Plugin (optional)

Setup Steps

Steps to setup the required development environment is listed

Setup Java Development Kit (JDK)

Download the latest version of Java JDK from Oracle's Java site: Java SE Downloads. Follow the given instructions to install and configure the setup. Finally set PATH and JAVA_HOME environment variables to refer to the directory that contains java and javac, typically java_install_dir/bin and java_install_dir respectively.

If running Windows and installed the JDK in C:\jdk1.6.0_15, put the following line in C:\autoexec.bat file.

```
set PATH=C:\jdk1.6.0_15\bin;%PATH%
set JAVA_HOME=C:\jdk1.6.0_15
```

Alternatively, right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, update the PATH value and press the OK button.

On Linux, if the SDK is installed in /usr/local/jdk1.6.0_15 and you use the C shell, put the following code into .cshrc file.

```
setenv PATH /usr/local/jdk1.6.0_15/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.6.0_15
```

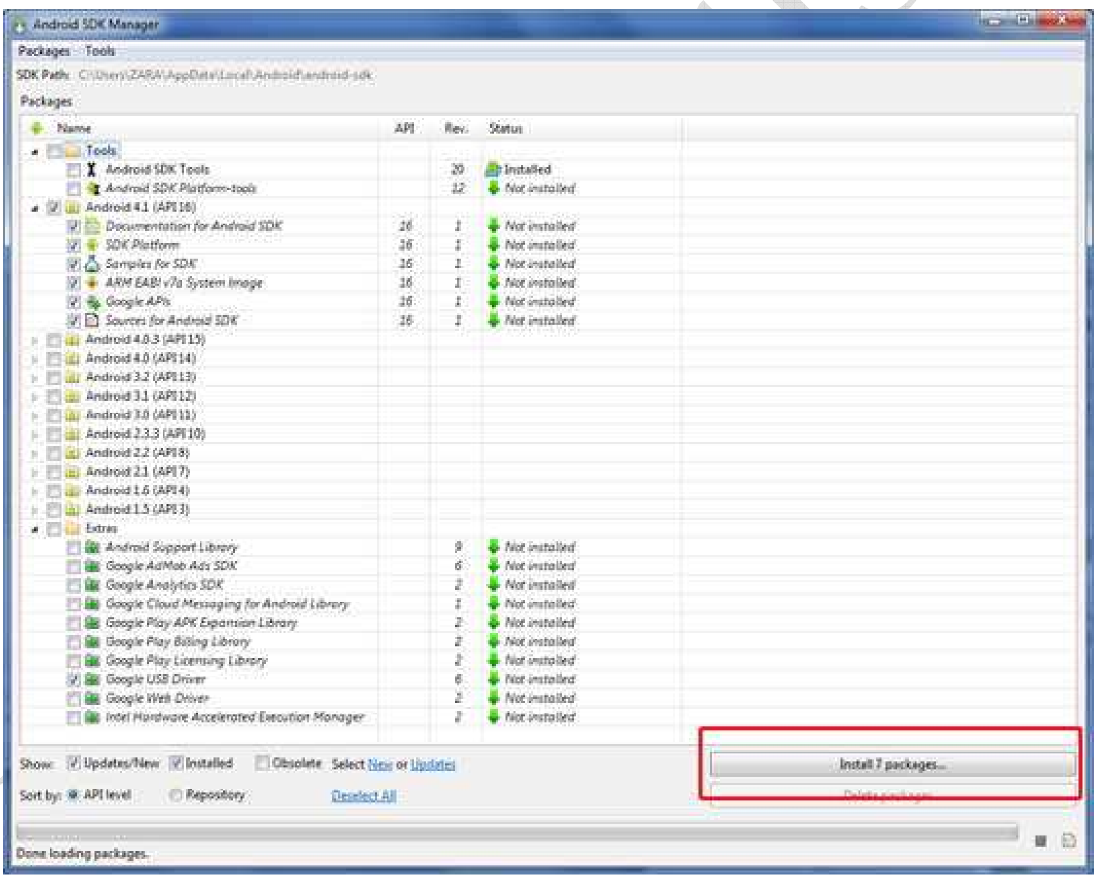
Alternatively, if using an Integrated Development Environment (IDE) Eclipse, then it will know automatically where Java is installed.

Setup Android SDK

Download the latest version of Android SDK from Android official website. If installing SDK on Windows machine, then a installer_rXX-windows.exe is to be downloaded, so just download and run this exe which will launch Android SDK Tool Setup wizard guides throughout of the installation, so just follow the instructions carefully.

If installing SDK either on Mac OS or Linux, check the instructions provided along with the downloaded android-sdk_rXX-macosx.zip file for Mac OS and android-sdk_rXX-linux.tgz file for Linux. Environment setup for Windows machine is listed below.

Launch Android SDK Manager using the option All Programs > Android SDK Tools > SDK Manager, this will give you following window



SDK Path: C:\Users\ZARA\AppData\Local\Android\android-sdk

Packages

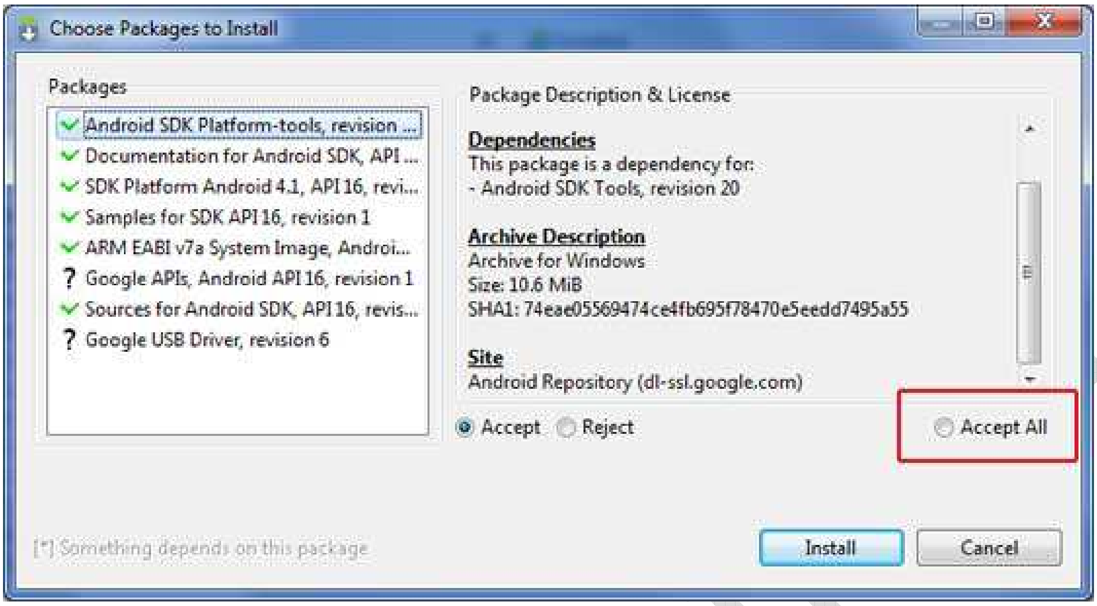
Name	API	Rev.	Status
□ Tools			
□ Android SDK Tools		20	Installed
□ Android SDK Platform-tools		12	Not installed
☑ Android 4.1 (API 16)			
☑ Documentation for Android SDK	16	1	Not installed
☑ SDK Platform	16	1	Not installed
☑ Samples for SDK	16	1	Not installed
☑ ARM EABI v7a System Image	16	1	Not installed
☑ Google APIs	16	1	Not installed
☑ Sources for Android SDK	16	1	Not installed
□ Android 4.0.3 (API 15)			
□ Android 4.0 (API 14)			
□ Android 3.2 (API 13)			
□ Android 3.1 (API 12)			
□ Android 3.0 (API 11)			
□ Android 2.3.3 (API 10)			
□ Android 2.2 (API 8)			
□ Android 2.1 (API 7)			
□ Android 1.6 (API 4)			
□ Android 1.5 (API 3)			
□ Extras			
□ Android Support Library		9	Not installed
□ Google AdMob Ads SDK		6	Not installed
□ Google Analytics SDK		2	Not installed
□ Google Cloud Messaging for Android Library		1	Not installed
□ Google Play APK Expansion Library		2	Not installed
□ Google Play Billing Library		2	Not installed
□ Google Play Licensing Library		2	Not installed
☑ Google USB Driver		6	Not installed
□ Google Web Driver		2	Not installed
□ Intel Hardware Accelerated Execution Manager		2	Not installed

Show: ☑ Updates/New ☑ Installed □ Obsolete Select New or Updates

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

Once you launched SDK manager, its time to install other required packages. By default it will list down total 7 packages to be installed, but I will suggest to de-select Documentation for Android SDK and Samples for SDK packages to reduce installation time. Next click Install 7 Packages button to proceed, which will display following dialogue box



Choose Packages to Install

Packages

- ✓ Android SDK Platform-tools, revision ...
- ✓ Documentation for Android SDK, API ...
- ✓ SDK Platform Android 4.1, API 16, revi...
- ✓ Samples for SDK API 16, revision 1
- ✓ ARM EABI v7a System Image, Androi...
- ? Google APIs, Android API 16, revision 1
- ✓ Sources for Android SDK, API 16, revis...
- ? Google USB Driver, revision 6

Package Description & License

Dependencies
This package is a dependency for:
- Android SDK Tools, revision 20

Archive Description
Archive for Windows
Size: 10.6 MiB
SHA1: 74eae05569474ce4fb695f78470e5eedd7495a55

Site
Android Repository (dl-ssl.google.com)

● Accept ○ Reject ○ Accept All

[*] Something depends on this package

If you agree to install all the packages, select Accept All radio button and proceed by clicking Install button. Now let SDK manager do its work and you go, pick up a cup of coffee and wait until all the packages are installed. It may take some time depending on internet connection. Once all the packages are installed, you can close SDK manager using top-right cross button.

Setup Eclipse IDE

All the examples in this tutorial have been written using Eclipse IDE. So I would suggest you should have latest version of Eclipse installed on machine.

To install Eclipse IDE, download the latest Eclipse binaries from http://www.eclipse.org/downloads/. Once you downloaded the installation, unpack the binary distribution into a convenient location. For example in C:\eclipse on windows, or /usr/local/eclipse on Linux and finally set PATH variable appropriately.

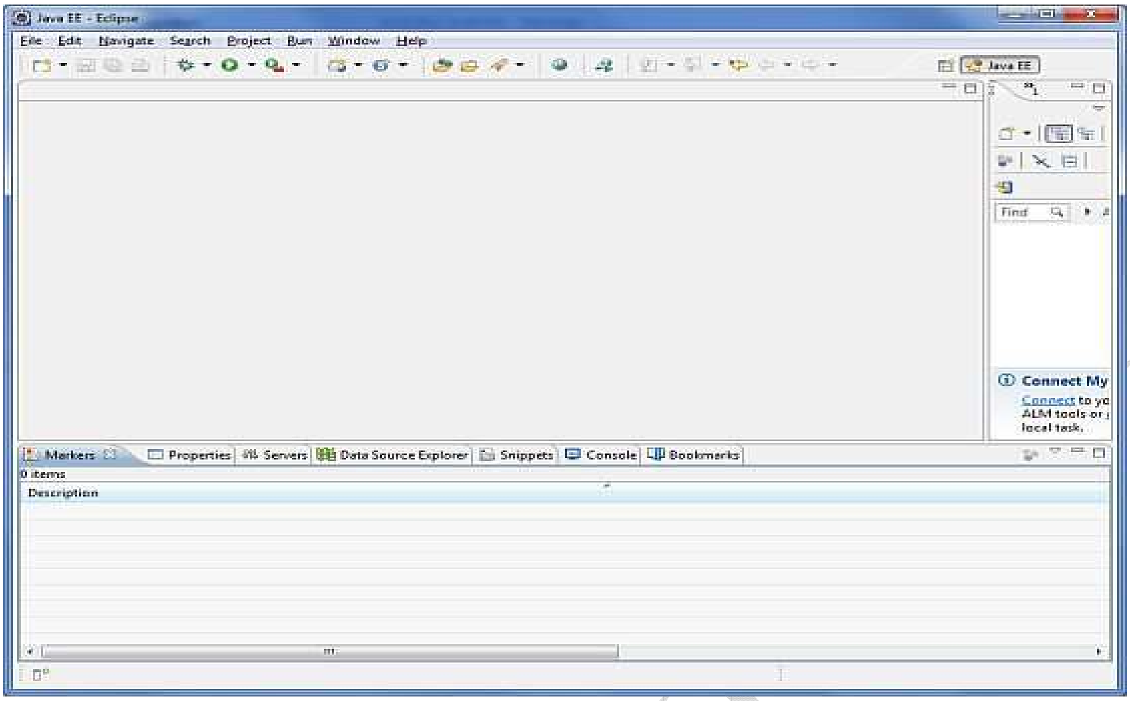
Eclipse can be started by executing the following commands on windows machine, or you can simply double click on eclipse.exe

```
 %C:\eclipse\eclipse.exe
```

Eclipse can be started by executing the following commands on Linux machine:

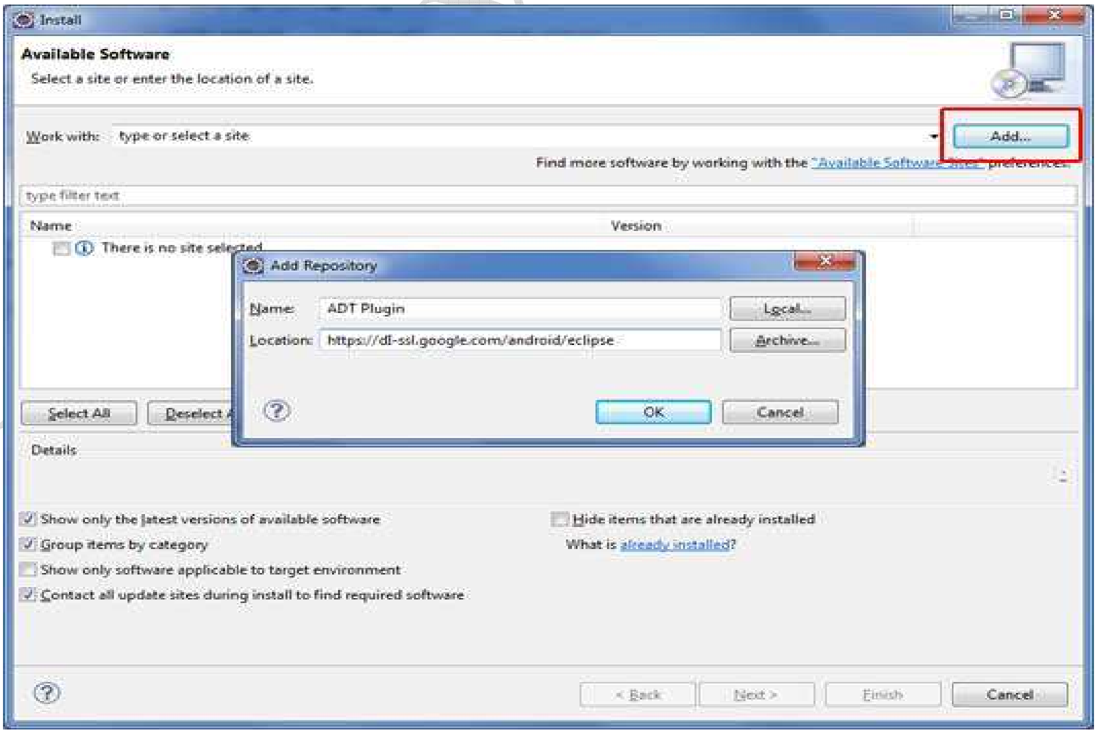
```
$/usr/local/eclipse/eclipse
```

After a successful startup, if everything is fine then it should display following result



Setup Android Development Tools (ADT) Plugin

This step will help you in setting Android Development Tool plugin for Eclipse. Let's start with launching Eclipse and then, choose Help > Software Updates > Install New Software. This will display the following dialogue box.


Install
Available Software
Select a site or enter the location of a site.
Work with: type or select a site
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
There is no site selected
Select All
Deselect All
Details
Add Repository
Local...
Archive...
OK
Cancel
< Back
Next >
Finish
Cancel

Name: ADT Plugin

Location: https://dl-ssl.google.com/android/eclipse

☑ Show only the latest versions of available software

☑ Group items by category

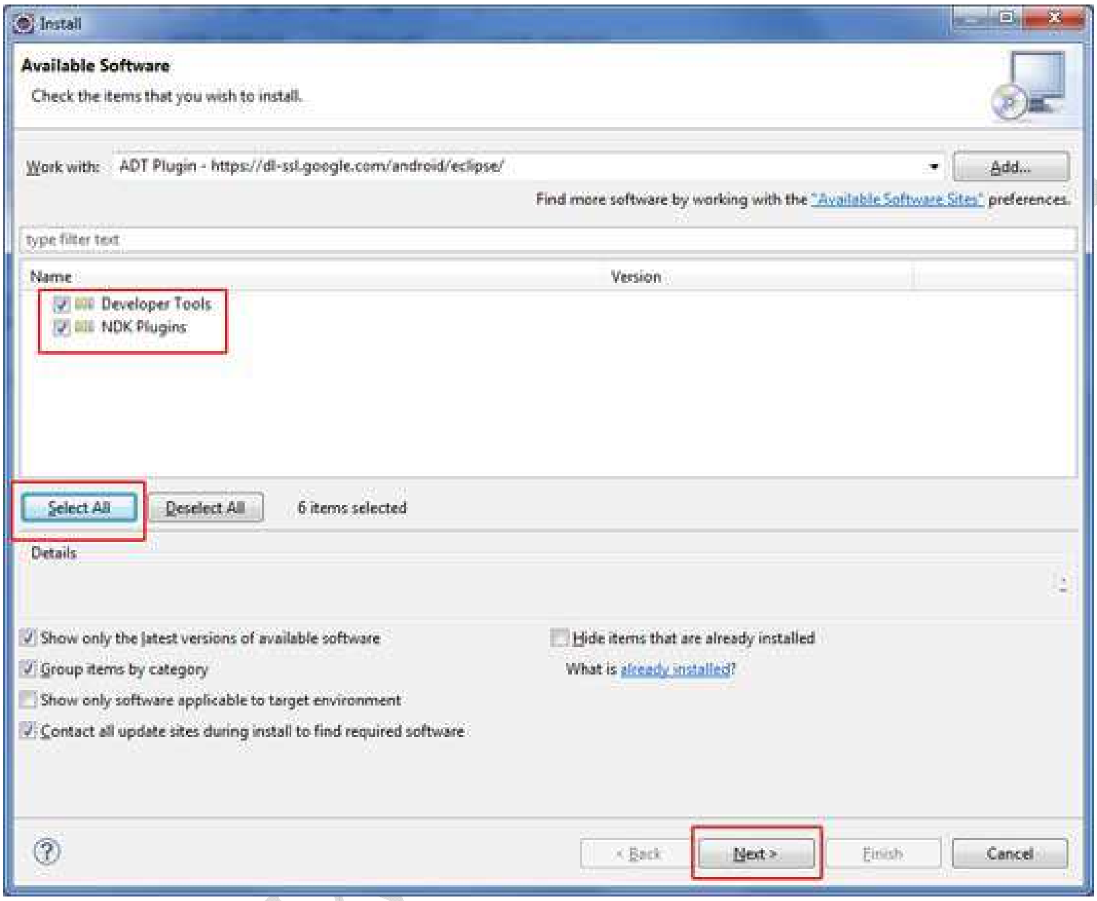
□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

□ Hide items that are already installed

What is already installed?

Now use Add button to add ADT Plugin as name and https://dl-ssl.google.com/android/eclipse/ as the location. Then click OK to add this location, as soon as you will click OK button to add this location, Eclipse starts searching for the plug-in available the given location and finally lists down the found plugins.



Install

Available Software

Check the items that you wish to install.

Work with: ADT Plugin - https://dl-ssl.google.com/android/eclipse/

Find more software by working with the "Available Software Sites" preferences.

type filter text

Name	Version
☑ Developer Tools	
☑ NDK Plugins	

6 items selected

Details

☑ Show only the latest versions of available software

☑ Group items by category

□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

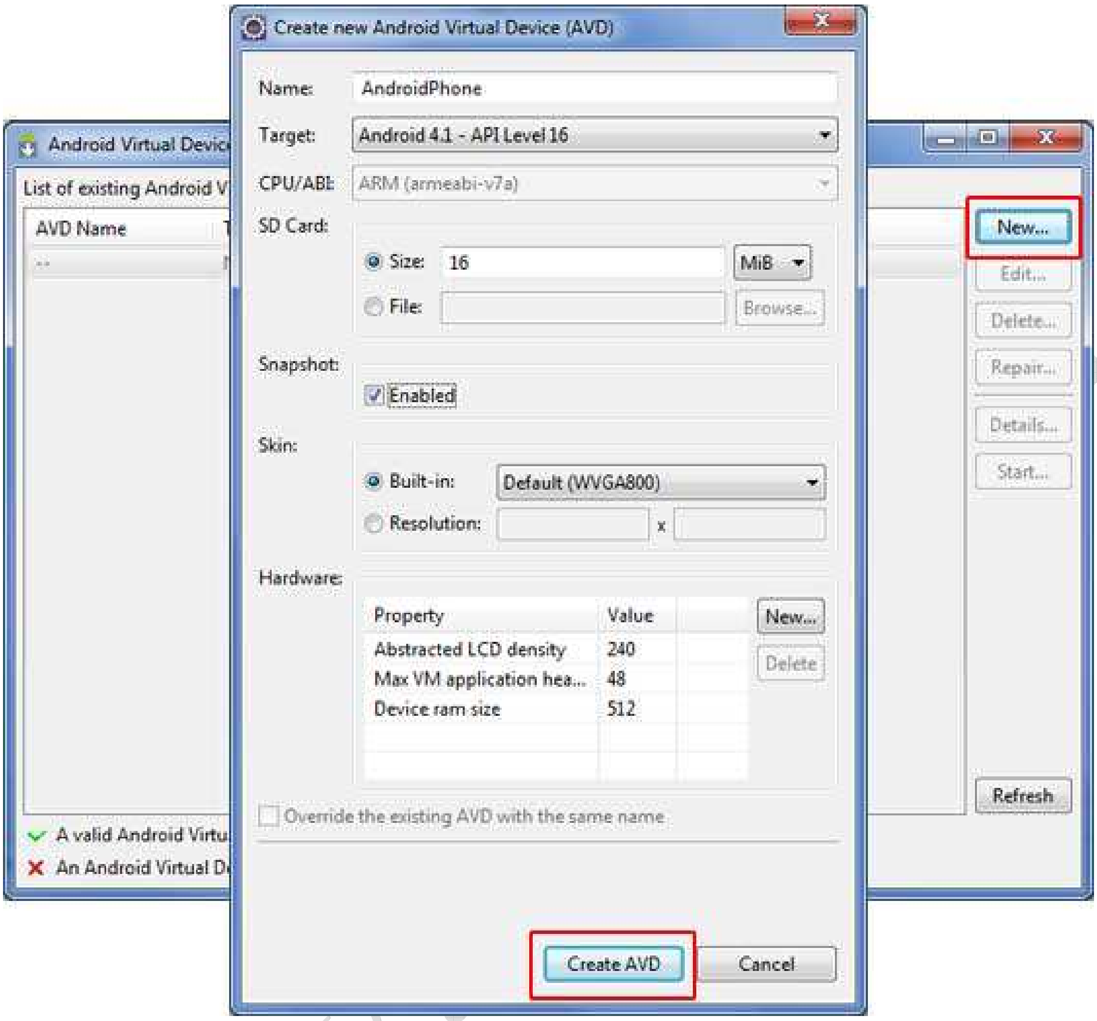
□ Hide items that are already installed

What is already installed?

Now select all the listed plug-ins using Select All button and click Next button which will guide you ahead to install Android Development Tools and other required plugins.

Create Android Virtual Device

To test Android applications you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager using Eclipse menu options Window > AVD Manager> which will launch Android AVD Manager. Use New button to create a new Android Virtual Device and enter the following information, before clicking Create AVD button.



Name: AndroidPhone

Target: Android 4.1 - API Level 16

CPU/ABI: ARM (armeabi-v7a)

SD Card:
● Size: 16 MiB
○ File:

Snapshot: ☑ Enabled

Skin:
● Built-in: Default (WVGA800)
○ Resolution: x

Hardware:

Property	Value
Abstracted LCD density	240
Max VM application hea...	48
Device ram size	512

□ Override the existing AVD with the same name

If AVD is created successfully it means environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re-start machine and once you are done with this last step, you are ready to proceed for first Android example but before that we will see few more important concepts related to Android Application Development.