UNIT – I
ANDROID AND DEVELOPMENT ENVIRONMENTS

VARIOUS MOBILE TECHNOLOGIES

Mobile technology is the technology used for cellular communication. . In wireless telecommunication major mobile technologies are CDMA, GSM (2G, 3G, 4G, 5G etc), GPRS, EDGE, UMTS, LTS etc

GSM

GSM(Global System for Mobile communication) is the popular mobile network which is used in maximum parts of the World. It is developed by The European Telecommunications Standards Institute and is used by 2 Billion People (approx.) around the World. It is under 2G mobile network connectivity and mainly depends on 900 MHz and 1900 MHz network bands. It also supports 3G Technology across World. Some of the GSM network providers in India are Airtel, Idea, Vodafone, Tata Docomo and BSNL etc.
Some of the Features/Advantages of GSM Network are

- Voice Calls
- Call Forwarding
- Call Barring
- Call waiting
- SMS – Short Message Service
- Voice Conference

1G

- This was the first generation of GSM, and it was an analog technology.
- Designed exclusively for voice communication
- Introduced in US in early 1980s
- speed upto 2.4kps
- Mobile phones with limited battery life
- No datasecurity

2G

- 2G stands for “second generation.
- 2G was digital (rather than analog)
- Introduced in Finland in 1991
- 2G introduced encryption
- There were GSM and CDMA versions of 2G

- Speed upto 64kbps
- Text and multimedia messaging possible
- 2G with GPRS is 2.5G

3G

- Speed up to 200 kbit/sec, and later versions could achieve multiple megabits per second.
- High speed browsing
- Suppots video conferencing, multimedia emails etc.
- Fast and easy audio,video transfer
- 3D gaming

HSDPA/HSPA

HSDPA(High Speed Downlink Packet Access) is an advanced technology to 3G Technology i.e., 3.5 Technology. It supports a speed of 7.2 Mbps (Megabits per Second) but its actual speed is 3 Mbps only. It supports to load larger files, MobileTV Streaming and Road Maps etc.

HSUPA – High Speed Uplink Packet Access is another technology besides of HSDPA. It is created by Nokia and supports a speed of 5.76 Mbps (Megabits per Second).

HSDPA and HSUPA together called as HSPA.

UMTS

UMTS(Universal Mobile Telecommunications System) is also a 3G Technology i.e., 3rd Generation Technology which is commonly called as WCDMA (Wideband CDMA). It provides faster data transfer rates at 42 Mbps (Megabits per second).

EV-DO

EV-DO(Evolution Data-Only) mainly runs on CDMA Networks for 3G. It supports a speed of 2.4 Mbps (Megabits per Second) but its actual speed is 450 Kbps(Kilobits per Second).

4G

- The major advance of 4G is mobile broadband internet services provided to external systems, such as laptops, wireless modems, etc.
- 4G introduced in 2011
- Speed 100Mbps to 1Gbps
- HD mobile TV
- Cloud computing
- IP telephony

LTE

LTE(Long Term Evolution) is a 4G Technology developed for GSM Network. It is the first 4G Technology used in mobile phones across world and was proposed by NTT DoCoMo. It is a high speed data transferor for mobile phones with 299.6 Mbps (Megabits per Second) download speed and 75.4 Mbps up load speed

CDMA

CDMA(Code Division Multiple Access) is a 3G wireless technology, which competes with GSM network and it is also used in different parts of world. Some of the key Features/Advantages of CDMA Network are
Along with GSM features, it provides

- Good Signal quality
- Voice Clarity
- Minimizes signal break up
- More reliable Network

VARIOUS MOBILE OPERATING SYSTEMS

A mobile operating system, also called a mobile OS, is an operating system that is specifically designed to run on mobile devices such as mobile phones, smartphones, PDAs, tablet computers and other handheld devices.

Types of Mobile Operating Systems

1. **Android OS (Google Inc.)**

The Android mobile operating system is Google's open and free software stack that includes an operating system, middleware and also key applications for use on mobile devices, including smartphones.

2. **Bada (Samsung Electronics)**

Bada is a proprietary Samsung mobile OS that was first launched in 2010. The Samsung Wave was the first smartphone to use this mobile OS. Bada provides mobile features such as multipoint-touch, 3D graphics and of course, application downloads and installation.

3. **BlackBerry OS (Research In Motion)**

The BlackBerry OS is a proprietary mobile operating system developed by Research In Motion for use on the company’s popular BlackBerry handheld devices.

4. **iPhone OS / iOS (Apple)**

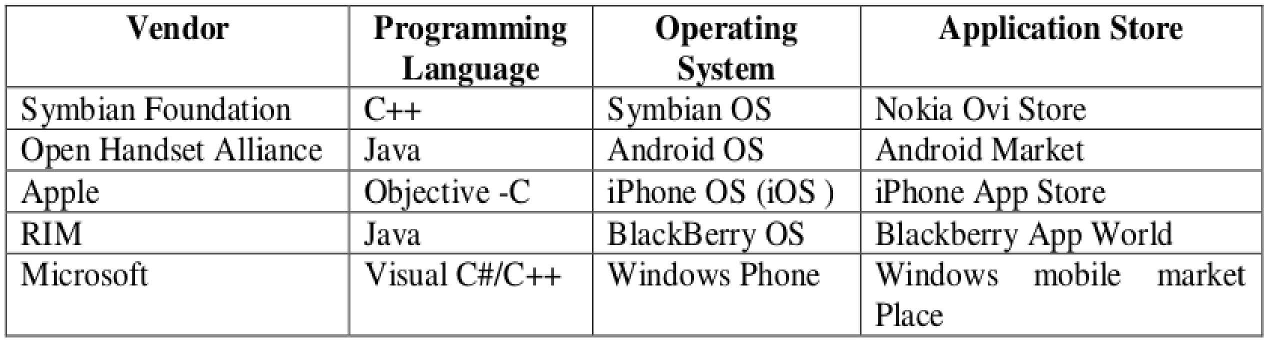
Apple's iPhone OS was originally developed for use on its iPhone devices. Now, the mobile operating system is referred to as iOS and is supported on a number of Apple devices including the iPhone, iPad, iPad 2 and iPod Touch. The iOS mobile operating system is available only on Apple's own manufactured devices

5. **Symbian OS (Nokia)**

Symbian is a mobile operating system (OS) targeted at mobile phones that offers a high-level of integration with communication and personal information management (PIM) functionality.

6. **Windows Mobile (Windows Phone)**

Windows Mobile is Microsoft's mobile operating system used in smartphones and mobile devices – with or without touchscreens. In 2010 Microsoft announced a new smartphone platform called Windows Phone 7.



Vendor	Programming Language	Operating System	Application Store
Symbian Foundation	C++	Symbian OS	Nokia Ovi Store
Open Handset Alliance	Java	Android OS	Android Market
Apple	Objective -C	iPhone OS (iOS)	iPhone App Store
RIM	Java	BlackBerry OS	Blackberry App World
Microsoft	Visual C#/C++	Windows Phone	Windows mobile market Place

Fig: Features of some mobile platforms

Apple IOS

- iPhone OS is a mobile operating system
- Developed by Apple Inc. exclusively for its hardware.
- Used in mobile devices, including the iPhone, iPad, and iPod Touch.
- It is the second most popular mobile operating system globally after Android.
- Apple's App Store contains iOS applications
- The iOS user interface is based upon direct manipulation, using multi-touch gestures.
- Interface control elements consist of sliders, switches, and buttons.
- Interaction with the OS includes gestures such as swipe, tap, pinch, and reverse pinch etc
- Internal accelerometers are used by some applications to respond to shaking the device
- Switching between portrait and landscape mode
- Incorporating thorough accessibility functions into iOS, enabling users with vision and hearing disabilities to properly use its products.
- The current version, iOS 11, was released on September 19, 2017

Android operating system

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Features of Android

- **Messaging**
  SMS and MMS are available forms of messaging, including threaded text messaging and Android Cloud To Device Messaging (C2DM)

- **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.

- **Voice-based features**
  Google search through voice has been available since initial release.[4] Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards.
- **Multi-touch**
  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.

- **Multitasking**
  Multitasking of applications, with unique handling of memory allocation, is available.

- **Screen capture**
  Android supports capturing a screenshot by pressing the power and home-screen buttons at the same time.

- **TV recording**
  Android TV supports capturing video and replaying it.

- **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.

- **Multiple language support**
  Android supports multiple languages

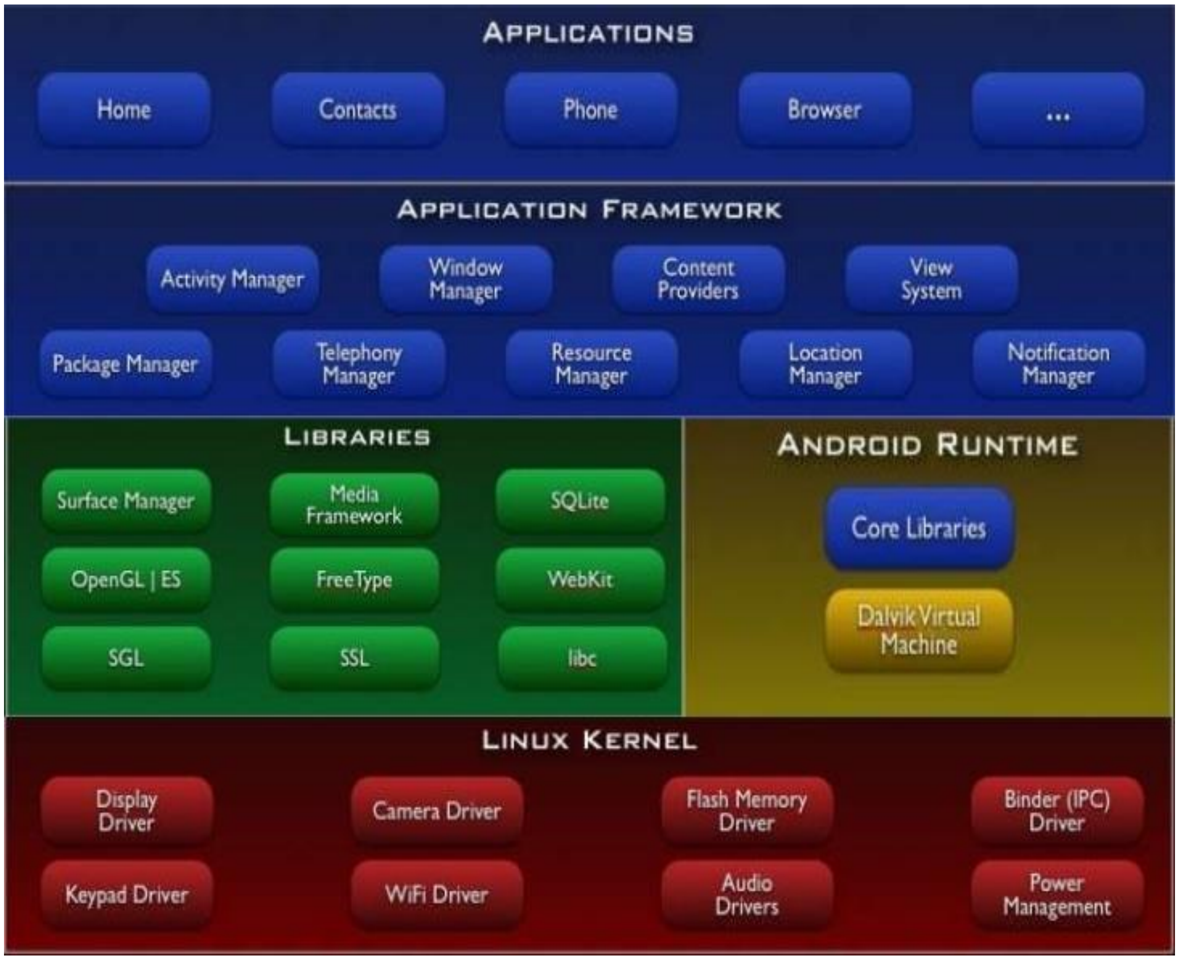
- **Accessibility**
  Built-in text-to-speech is provided by TalkBack for people with low or no vision.

- **Storage**
  SQLite, a lightweight relational database, is used for data storage purposes.

- **Wi-Fi Direct**
  A technology that let apps discover and pair directly, over a high-bandwidth peer-to-peer connection.

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.



1. 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.

2. Libraries

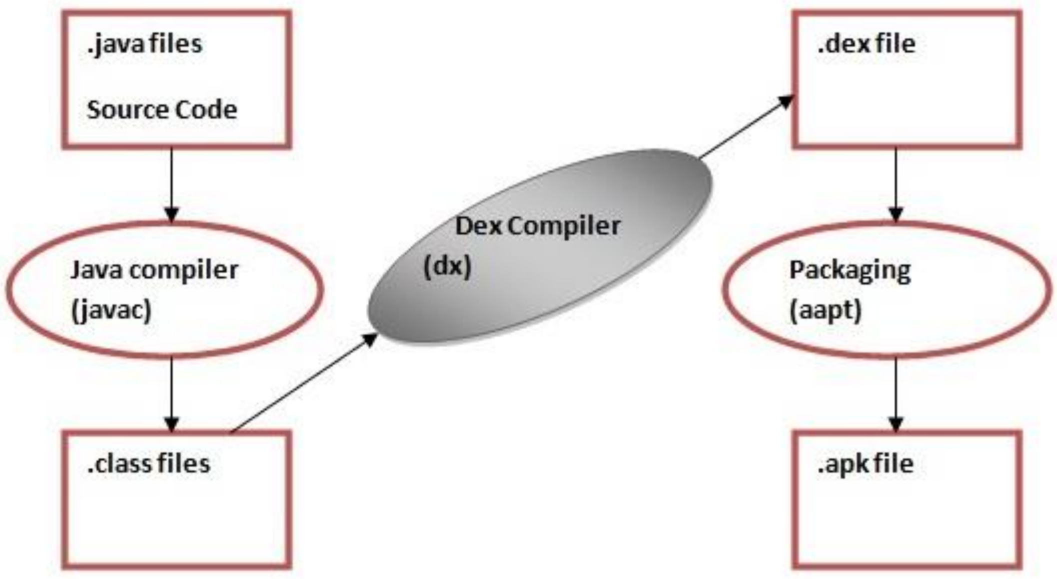
On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security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

3. 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 Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Dalvik Virtual Machine | DVM

The **Dalvik Virtual Machine (DVM)** is an android virtual machine optimized for mobile devices. It optimizes the virtual machine for memory, battery life and performance. The Dex compiler converts the class files into the .dex file that run on the Dalvik VM. Multiple class files are converted into one dex file. Below figure shows the compiling and packaging process from the source file:



The **javac tool** compiles the java source file into the class file.
The **dx tool** takes all the class files of your application and generates a single .dex file. It is a platform-specific tool.
The **Android Assets Packaging Tool (aapt)** handles the packaging process.

4. 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.

ANDROID APPLICATION COMPONENTS

The **core building blocks** or **fundamental components** of android are activities, views, intents, services, content providers, fragments and AndroidManifest.xml.

- **Activity**
  An activity is a class that represents a single screen. It is like a Frame in AWT.

- **View**
  A view is the UI element such as button, label, text field etc. Anything that you see is a view.

- **Intent**

Intent is used to invoke components. It is mainly used to:

  - Start the service
  - Launch an activity
  - Display a web page
  - Display a list of contacts
  - Broadcast a message
  - Dial a phone call etc.
  -

For example, you may write the following code to view the webpage.

```
Intent intent=new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("http://www.javatpoint.com"));
startActivity(intent);
```

- **Service**

Service is a background process that can run for a long time. There are two types of services local and remote. Local service is accessed from within the application whereas remote service is accessed remotely from other applications running on the same device.

- **Content Provider**

Content Providers are used to share data between the applications.

- **Fragment**

Fragments are like parts of activity. An activity can display one or more fragments on the screen at the same time.

- **AndroidManifest.xml**

It contains informations about activities, content providers, permissions etc. It is like the web.xml file in Java EE.

- **Android Virtual Device (AVD)**

It is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.

- **Android Emulator**

Android Emulator is used to run, debug and test the android application. If you don't have the real device, it can be the best way to run, debug and test the application.
It uses an open source processor emulator technology called QEMU.

The emulator tool enables you to start the emulator from the command line. You need to write:

```
emulator -avd <AVD NAME>
```

In case of Eclipse IDE, you can create AVD by Window menu > AVD Manager > New.

COMPARISON BETWEEN APPLE IOS AND ANDROID

	Android	**iOS**
Customizability	A lot. Can change almost anything.	Limited unless jailbroken
Developer	Google, Open Handset Alliance	Apple Inc.
Initial release	September 23, 2008	July 29, 2007
Source model	Open source	Closed, with open source components.
OS family	Linux	OS X, UNIX
Widgets	Yes	No, except in NotificationCenter
File transfer	Easier than iOS.	More difficult. Media files can be transferred using iTunes desktop app.
Available on	Many phones and tablets. Major manufacturers are Samsung, Motorola, LG, HTC and Sony.. Nexus and Pixel line of devices is pure Android, others bundle manufacturer software.	iPod Touch, iPhone, iPad, Apple TV (2nd and 3rd generation)
Calls and messaging	Google Hangouts. 3rd party apps like Facebook Messenger, WhatsApp, Google Duo and Skype all work on Android and iOS both.	iMessage, FaceTime (with other Apple devices only). 3rd party apps like Google Hangouts, Facebook Messenger, WhatsApp, Google Duo and Skype all work on Android and iOS both.
Internet browsing	Google Chrome (or Android Browser on older versions; other browsers are available)	Mobile Safari (Other browsers are available)
App store ,	Google Play – 1,000,000+	Apple app store – 1,000,000+ apps

Affordability and interface	apps. Other app stores like Amazon and Getjar also distribute Android apps. (unconfirmed ".APKs")	
Available language(s)	100+ Languages	34 Languages
Video chat	Google Duo and other 3rd party apps	FaceTime (Apple devices only) and other 3rd party apps
Voice commands	Google Now, Google Assistant	Siri
Maps	Google Maps	Apple Maps (Google Maps also available via a separate app download)
Latest stable release and Updates	Android 8.0.0, Oreo (Aug 21, 2017)	11 (Sep 19, 2017)
Alternative app stores and side loading	Several alternative app stores other than the official Google Play Store. (e.g. Aptoide, Galaxy Apps)	Apple blocks 3rd party app stores. The phone needs to be jailbroken if you want to download apps from other stores.
Battery life and management	Many Android phone manufacturers equip their devices with large batteries with a longer life.	Apple batteries are generally not as big as the largest Android batteries. However, Apple is able to squeeze decent battery life via hardware/software optimizations.
Open source	Kernel, UI, and some standard apps	The iOS kernel is not open source but is based on the open-source Darwin OS.
File manager	Yes. (Stock Android File Manager included on devices running Android 7.1.1)	Not available
Photos & Videos backup	Apps available for automatic backup of photos and videos. Google Photos allows unlimited backup of photos. OneDrive, Amazon Photos and Dropbox are other alternatives.	Up to 5 GB of photos and videos can be automatically back up with iCloud. All other vendors like Google, Amazon, Dropbox, Flickr and Microsoft have auto-backup apps for both iOS and Android.

Security	Android software patches are available soonest to Nexus device users. Manufacturers tend to lag behind in pushing out these updates. So at any given time a vast majority of Android devices are not running updated fully patched software.	Most people will never encounter a problem with malware because they don't go outside the Play Store for apps. Apple's software updates support older iOS devices also.
Rooting, bootloaders, and jailbreaking	Access and complete control over your device is available and you can unlock the bootloader.	Complete control over your device is not available.
Cloud services	Native integration with Google cloud storage. 15GB free, $2/mo for 100GB, 1TB for $10. Apps available for Amazon Photos, OneDrive and Dropbox.	Native integration with iCloud. 5GB free, 50GB for $1/mo, 200GB for $3/mo, 1TB for $10/mo. Apps available for Google Drive and Google Photos, Amazon Photos, OneDrive and Dropbox.
Working state	Current	Current
Interface	Touch Screen	Touch Screen
Supported versions	Android 5.0 & later (Android 4.4 is also supported but with patches)	iOS 8 & later
First version	Android 1.0, Alpha	iOS 1.0

Different android supporting devices

- ➢Smartphones
- ➢Tablets
- ➢E-reader devices
- ➢Netbooks
- ➢MP4 players
- ➢Internet Tvs

Different IDE for Android application development

- Eclipse
- Android SDK – Android studio
- Android Development Tools (ADT)

Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. Eclipse is written mostly in Java and its primary use is for developing Java applications

Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system designed specifically for Android development
Android Studio was announced on May 16, 2013 at the Google I/O conference

Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

Android Development Tools (ADT)

It is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications. ADT extends the capabilities of Eclipse to let you quickly set up new Android projects, create an application UI, add packages based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) .apk files in order to distribute your application.

INSTALLING ECLIPSE IN LINUX

Before proceeding, make sure that we have JDK is installed.

1. Install the Java Development Kit (JDK)

   Step1 : $ sudo add-apt-repository ppa:webupd8team/java
   Step2: $ sudo apt-get update
   Step 3: $ sudo apt-get install oracle-java8-installer oracle-java8-set-default
   Verify if java installation was successful or not by checking the version number
   Step4: : $ java -version

2. Download Eclipse IDE from its website.
   Check out your OS Type, 32-bit or 64-bit, and select Linux 32-bit or 64-bit of Eclipse Installer.
3. Run the installer wizard.
   Decompress the downloaded archive -
   $tar xf eclipse-inst-linux64.tar.gz

Run eclipse-inst -

```
$./eclipse-inst
```

When the wizard launches, select install item and folder, and finally click INSTALL button.

4. Download the ADT plugin for eclipse
5. Configuring the ADT plugin
6. Create an Android Virtual Device (AVD)
7. Create and run the simple android example

INSTALLING ANDROID STUDIO IN LINUX

Step1 : Install JDK 1.8.

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer oracle-java8-set-default
```

Verify if java installation was successful or not by checking the version number

```
$ java –version
```

Step2: `$ sudo apt-get update`

Step3: `$ sudo apt-get install android-studio`

Above install command will install android-studio in the directory /opt.

(or download the zip file and uncompress into /opt)

Now, run the following command to start the setup wizard:

Step4: `$ /opt/android-studio/bin/studio.sh`

This will invoke the setup screen.

- Select standard type installation
- Once you press the Finish button, Licence agreement will be displayed.
- After you accept the licence, it starts downloading the required components.
- Android studio installation will be complete after this step.
- On completion, launch the android studio and configure SDK Manager and AVD
- When you relaunch Android studio, you will be shown welcome screen from where you will be able to start working with your Android Studio- create and run the simple android example

INSTALLING ANDROID STUDIO IN WINDOWS

Android programming is Java based and it requires JDK & JRE environment. After installing JDK & JRE, configure environment variables.

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

The next step will be installation of Android IDE. After downloading android studio software,

Step1: Open you downloaded android studio file
Step2: It will show you welcome to android studio. - Go for "Next"
Step3: Now choose to install, android programming. - Go for "Next"
Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation.
Step6: Now choose the location for Android Studio and SDK. - Go for "Next"
Step7: Here you need to define the size of android emulator processor. You can define up to 2GB and more than that, depends on your RAM capacity. This will help your android studio to run faster.
Step8: Now choose setting for shortcut of android studio on your computer.
Step9: After that it will copy all the files to the storage location you have selected before.
Step10. On completion, launch the android studio and configure SDK Manager and AVD
Step11. Create and run the simple android example

INSTALLING ANDROID SDK MANAGER

To download and install latest android APIs and development tools from the internet, android provide us with android SDK manager. Android SDK Manager separates the APIs, tools and different platforms into different packages.

- launch Android SDK Manager - -
  - You can select which package you want to download by selecting the checkbox and then click **Install** to install those packages.
  - By default SDK Manager keeps it up to date with latest APIs and other packages. Gradle can automatically download missing SDK packages that a project depends on.
  - Once you download the SDK, following packages are available but first three are necessary to run your SDK and others are recommended.

SNo	Package & Description
1	**SDK Build-Tools -** It includes tools to build Android apps.
2	SDK Platform-tools - Includes various tools required by the Android platform, including the adb tool.
3	SDK Platform - At least one platform must be installed in your environment to run your application.
4	System Image - It's a good practice to download system images for all of the android versions so you can test your app on them with the Android Emulator.

ANDROID VIRTUAL DEVICE

An Android Virtual Device (AVD) is an emulator configuration that allows developers to test the application by simulating the real device capabilities. We can configure the AVD by specifying the hardware and software options. AVD manager enables an easy way of creating and managing the AVD with its graphical interface. We can create as many AVDs as we need, based on the types of device we want to test for.. This emulator to provide a virtual device-specific environment in which to install and run Android apps.

The Procedure For Creating Android Virtual Device

The AVD Manager is a tool you can use to create update, delete, repair, and manage Android virtual devices (AVDs), which define device configurations for the Android Emulator. To launch the AVD Manager:

- In Android Studio, select Tools > Android > AVD Manager, or click the AVD Manager icon in the toolbar->Create AVD
- Or, use the command line to navigate to your SDK's tools/ directory and execute:

  $ android avd
- In Eclipse ADT, Below are the steps to create an AVD from AVD manager graphical interface
  - Go to Window ->AVD Manager and select Virtual Devices.
  - Click on New to create a Virtual Device, give it some Name and select Target Android Platform from the drop down list
  - Click “Create AVD”

We can create as many AVDs as you would like to use with the Android Emulator. To effectively test our app, we should create an AVD that models each device type for which we have designed our app to support

How to create a Customize AVD for a new mobile model

In case the available device definitions do not match the device type you'd like to emulate, you can create a custom device definition for your AVD:

1. From the main screen, click Create Virtual Device.
2. To begin you custom device by using an existing device profile as a template,select a device profile then click Clone Device.
3. Or, to start from scratch, click New Hardware Profile.
4. The Configure Hardware Profile window allows you to specify various configurations such as the screen size, memory options, input type, and sensors.
5. When you're done configuring the device, click Finish.
6. Your custom device configuration is now available in the list of device definitions

Hardware options for creating a new AVD

Characteristic	Description	Property
Device ram size	The amount of physical RAM on the device, in megabytes. Default value is "96".	hw.ramSize
Touch-screen support	Whether there is a touch screen or not on the device. Default value is "yes".	hw.touchScreen
Trackball support	Whether there is a trackball on the device. Default value is "yes".	hw.trackBall
Keyboard support	Whether the device has a QWERTY keyboard. Default value is "yes".	hw.keyboard
DPad support	Whether the device has DPad keys. Default value is "yes".	hw.dPad
GSM modem support	Whether there is a GSM modem in the device. Default value is "yes".	hw.gsmModem
Camera support	Whether the device has a camera. Default value is "no".	hw.camera
Maximum horizontal camera	Default value is "640".	hw.camera.maxHorizontalPixels

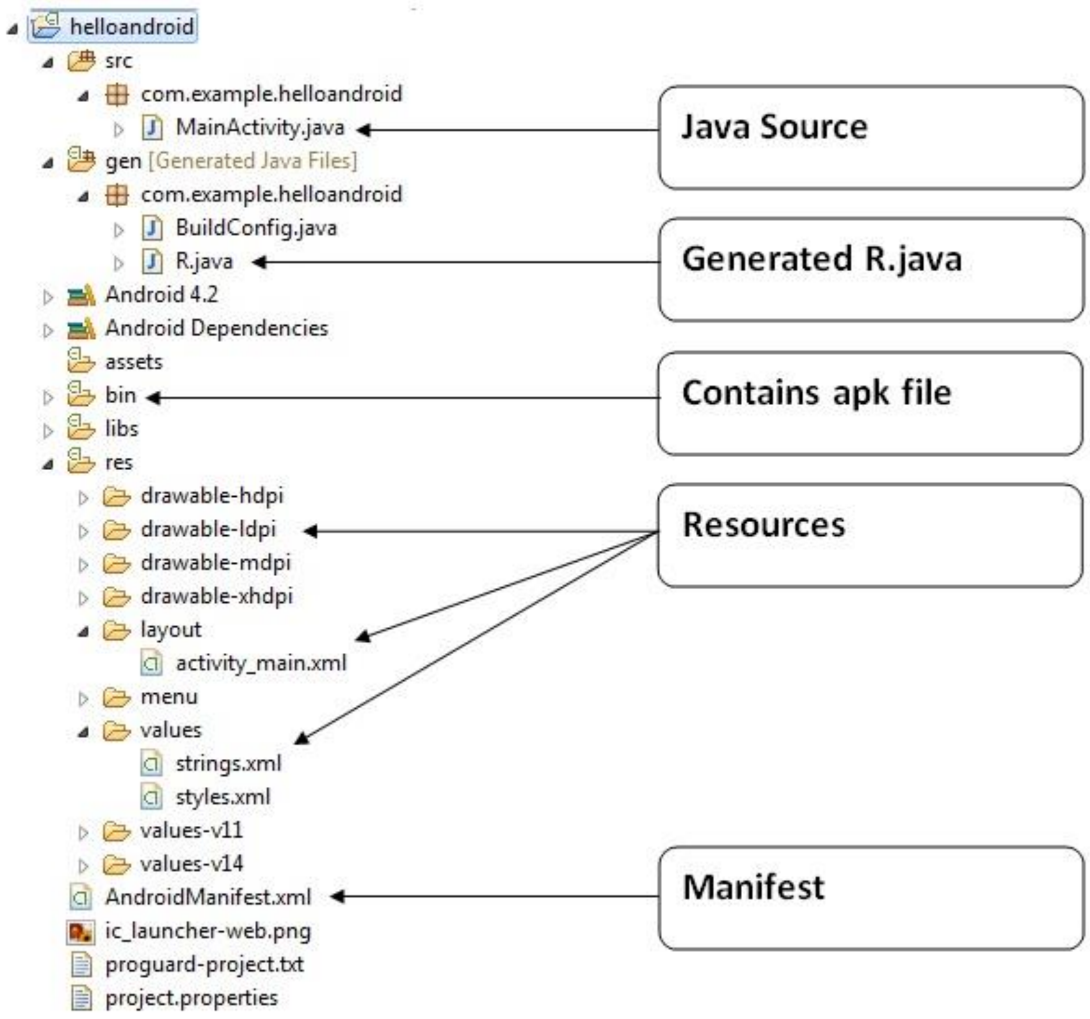
pixels		
Maximum vertical camera pixels	Default value is "480".	hw.camera.maxVerticalPixels
GPS support	Whether there is a GPS in the device. Default value is "yes".	hw.gps
Battery support	Whether the device can run on a battery. Default value is "yes".	hw.battery
Accelerometer	Whether there is an accelerometer in the device. Default value is "yes".	hw.accelerometer
Audio recording support	Whether the device can record audio. Default value is "yes".	hw.audioInput
Audio playback support	Whether the device can play audio. Default value is "yes".	hw.audioOutput
SD Card support	Whether the device supports insertion/removal of virtual SD Cards. Default value is "yes".	hw.sdCard
Cache partition support	Whether we use a /cache partition on the device. Default value is "yes".	disk.cachePartition
Cache partition size	Default value is "66MB".	disk.cachePartition.size
Abstracted LCD density	Sets the generalized density characteristic used by the AVD's screen. Default value is "160".hw.lcd.density	hw.lcd.density

CREATING A NEW PROJECT ON ANDROID STUDIO

1. Start your Android Studio. It will show you the default start window
2. Click on Start a new Android Studio project.
3. Provide project information or configuration.
   - put Application Name - Without Space, Capitalized letter format
   - Company Domain,
   - Package Name,
   - Project Location,
   - choose platform where you want to run the android application
   - choose the activity – Blank Activity, you will get auto generated codes
   - select layout name, activity name etc...
4. Click on Finish button

Internal Details of Hello Android Example

Android application contains different components such as java source code, string resources, images, manifest file, apk file etc. Let's understand the project structure of android application.



Various folders and their files created by an android project

- **src** — Contains the . java source files for your project. You will write the code for your application in this file.
- Android library — This item contains one file, android.jar , which contains all the class libraries needed for an Android application.
- **gen** — Contains the R.java file, a compiler-generated file that references all the resources found in your project.

You should not modify this file.

- **assets** — This folder contains all the assets used by your application, such as HTML, text files, databases, etc.
- **res** — This folder contains all the resources used in your application. It also contains a few other subfolders: drawable-<resolution> , layout , and values .
- **AndroidManifest.xml** — This is the manifest file for your Android application. Here you specify the permissions needed by your application, as well as other features (such as intent-filters,receivers,etc.).

Android project is now a basic "Hello World" app that contains the following default files.

- app/src/main/java/com.mycompany.myfirstapp/MainActivity.java – Class definition of the activity
- app/src/main/AndroidManifest.xml – Defines the components in the App
- app/src/main/res/layout/activity_main.xml - Activity seen in design mode and text mode

File : MainActivity.java

```
package com.example.helloandroid;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
TextView textview=new TextView(this);
textview.setText("Hello Android!");
setContentView(textview);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.activity_main, menu);
return true;  } }
```

- **Activity** is a java class that creates and default window on the screen where we can place different components such as Button, EditText, TextView, Spinner etc. It is like the Frame of Java AWT. It provides life cycle methods for activity such as onCreate, onStop, OnResume etc.
- The **onCreate method** is called when Activity class is first created.

- The **setContentView(R.layout.activity_main)** gives information about our layout resource. Here, our layout resources are defined in activity_main.xml file.

File: activity_main.xml

```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"     xmln
s:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity" >
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_centerVertical="true"
      android:text="@string/hello_world" />
</RelativeLayout>
```

The message for this string is defined in the strings.xml file. The **@string/hello_world** provides information about the textview message. The value of the attribute hello_world is defined in the strings.xml file.

File: strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string name="app_name">helloandroid</string>
   <string name="hello_world">Hello world!</string>
   <string name="menu_settings">Settings</string>
</resources>
```

We can change the value of the hello_world attribute from this file.

Generated R.java file - It is the auto-generated file that contains IDs for all the resources of res directory. It is generated by aapt(Android Asset Packaging Tool). Whenever you create any component on activity_main, a corresponding ID is created in the R.java file which can be used in the Java Source file later.

APK File

An apk file is created by the framework automatically. If you want to run the android application on the mobile, transfer and install it.

Resources

It contains resource files including activity_main, strings, styles etc.

AndroidManifest.xml

The AndroidManifest.xml file contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.

It performs some other tasks also:

- It is responsible to protect the application to access any protected parts by providing the permissions.
- It also declares the android api that the application is going to use.
- It lists the instrumentation classes. The instrumentation classes provides profiling and other informations. These informations are removed just before the application is published etc.

This is the required xml file for all the android application and located inside the root directory.

A simple AndroidManifest.xml file is as follows:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.javatpoint.hello"
  android:versionCode="1"
  android:versionName="1.0" >
   <uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="15" />
   <application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
      android:name=".MainActivity"
      android:label="@string/title_activity_main" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    </application>
  </manifest>
```

Elements of the AndroidManifest.xml file

The elements used in the above xml file are described below.

- **<manifest> -** manifest is the root element of the AndroidManifest.xml file. It has package attribute that describes the package name of the activity class and also can specify the version code and version name of your application. xmlns specifies the xml namespace of a resource.
- **<uses sdk> -** Defines minimum sdk version and target sdk version
- **<application> -** application is the subelement of the manifest. It includes the namespace declaration. This element contains several subelements that declares the application component such as activity etc. The commonly used attributes are of this element are **icon**, **label**, **theme** etc.
  android:icon represents the icon for all the android application components.
  android:label works as the default label for all the application components.
  android:theme represents a common theme for all the android activities.

- **<activity> -** activity is the subelement of application and represents an activity that must be defined in the AndroidManifest.xml file. It has many attributes such as label, name, theme, launchMode etc.
  android:label represents a label i.e. displayed on the screen.
  android:name represents a name for the activity class. It is required attribute.

- **<intent-filter> -** intent-filter is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.
- **<action> -** It adds an action for the intent-filter. The intent-filter must have at least one action element.
- **<category> -** It adds a category name to an intent-filter.

****