Mobile Application Development

Year	Subject Title	Sub. Code
2018-19 Onwards	ELECTIVE-IV: MOBILE APPLICATION DEVELOPMENT	18MIT42E

- UNIT-I: Introduction to Android OS: Android Description – Open Handset Alliance – Android. Ecosystem – Android versions – Android Activity – Features of Android – Android Architecture – Stack Linux Kernel. Configuration of Android Environment: Operating System – Java JDK – Android SDK – Android Development Tools (ADT) – Android Virtual Devices (AVDs) – Emulators - Dalvik Virtual Machine – Differences between JVM and DVM – Steps to Install and Configure Eclipse and SDK. (Chapters 1 & 2)
- UNIT-II: Create the first android application: Directory Structure. Android User Interface: Understanding the Components of a screen– Linear Layout – Absolute Layout – Frame. Layout – Relative Layout – Table Layout. (Chapters 3 & 4)
- UNIT-III: Designing User Interface with View – Text View – Button – Image Button – Edit Text – Check Box – Toggle Button – Radio Button and Radio Group – Progress Bar – Auto complete Text View – Spinner – List View – Grid View – Image View - Scroll View – Custom Toast – Alert – Time and Date Picker. (Chapter 5)
- UNIT-IV: Activity: Introduction – Intent – Intent filter – Activity life cycle – Broadcast life cycle – Service. Multimedia: Android System Architecture – Play Audio and Video – Text to Speech. (Chapters 6 & 7)
- UNIT-V: SQLite Database in Android: SQLite Database – Creation and Connection of the database – Transactions. Case Study: SMS Telephony and Location Based Services. (Chapters 8, 9, & 10)

TEXT BOOK

1. Prasanna Kumar Dixit, "Android", Vikas Publishing House Private Ltd., Noida, 2014.

REFERENCE BOOKS

1. Reto Meier and Wrox Wiley, "Professional Android 4 Application Development", 2012.

2. ZiguradMednieks, LaridDornin, G.BlakeMeike, Masumi Nakamura, "Programming Andriod", O'Reilly,2013.

3. Robert Green, Mario Zechner, "Beginning Android 4 Games Development", Apress

Media LLC, New York, 2011

Introduction to Android operating system

- What is Android ?

- Android is a stack of software for mobile devices that are an operating system middleware and key applications

- Android is a Linux based operating system which is designed for touch screen mobile devices like Smartphones and tablet computers

- it is an open-source technology that allows the software to be freely modified and distributed by device manufacturers, wireless Carriers and developers
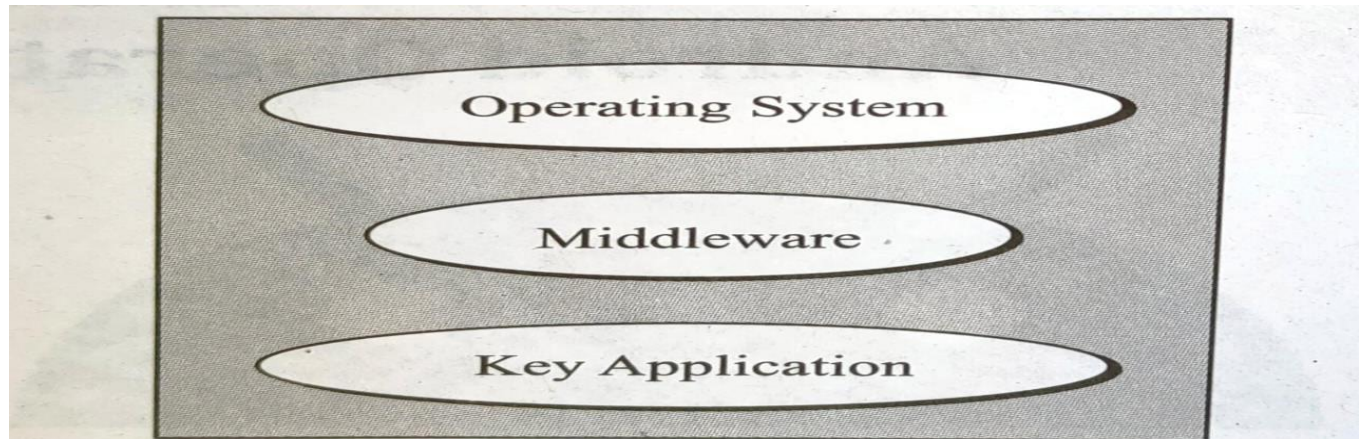

Figure 1.1: Android Operating System

What is "Open Handset Alliance" ?
OHA was formed in Nov 2007, created by Google, alliance of 84 firms, for developing open standard platform for mobile devices. The members of OHA include handset manufacturers, chip makers, commercialization companies, software companies and mobile operators
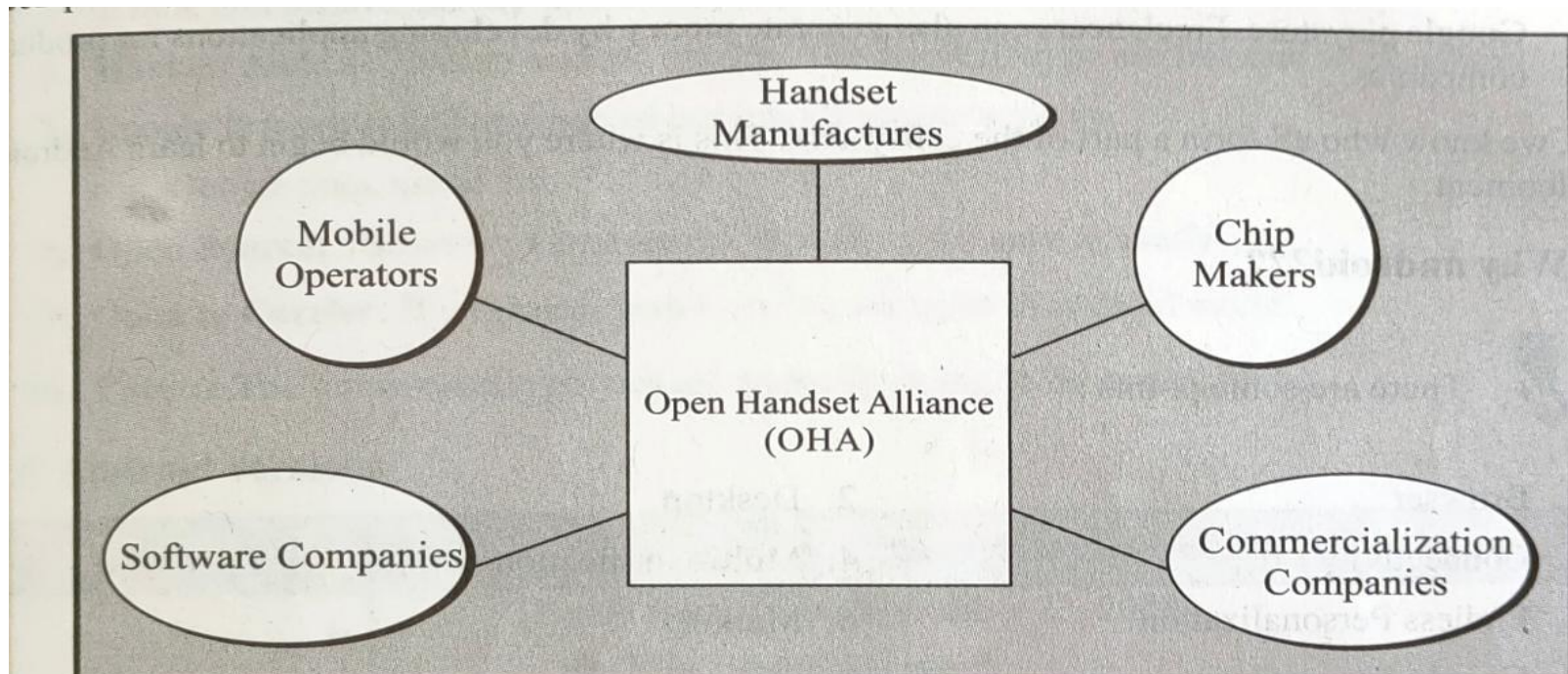

Figure 1.3 : Android Open Handset Alliance

- Android Ecosystem
- Why Android ?
- Android versions
- Android Activity
- Features of Android
- Android Architecture
- Stack Linux Kernel

Android Versions

Versions	APK/API level	Release date and Activity
1.0	1	Released on Sep 23, 2008 -You tube video player - Wi-fi & Bluetooth support -Camera
1.1 Petit Four (only internally used)	2	Feb 9, 2009 -Ability to show/hide dial pad -Ability to show attachments in messages
1.5 Cup cake	3	Apr 27, 2009 -Recording and watching videos in MPEG-4 and3GP formats -Home screen with widgets
1.6 Donut	4	-Sep 15, 2009 -Voice & text entry search enhanced -Multi-lingual speech synthesis engine -Gallery, camera & camcoder integrated -Ability to select multiple photos -Support for WVGA screen resolutions

Versions	APK/API level	Release date and Activity
2.0 Eclairs	5	Oct 26, 2009 - allowing users to add multiple accounts to a device for synchronisation of email and contacts -Bluetooth 2.1 support -ability to tap contacts photo and select call, SMS or e-mail -ability to search all saved SMS and MMS messages -various camera features like flash support, digital zoom, colour effect -improved Google Maps -live wallpapers → animation of home screen background images
2.0.1	6	Dec 3 2009 -minor API changes, bug fixes
2.1	7	Jan 12 2010 -minor API changes, bug fixes
2.2 Froyo	8	May 20, 2010 -speed, memory and performance optimization -speed improvements through JIT compilation (just in time) -support for the Android cloud to device messaging service (c2dm) -USB tethering and Wi-Fi hotspot functionality

Versions	APK/API level	Release date and Activity
2.3 Gingerbread	9	Dec 6, 2010 -support for extra large screen sizes and resolutions (WXGA and higher) -native support for SIP VOIP internet telephones -enhanced copy/ paste functionality -support for near field communication (NFC), to read NFC tag in a poster or sticker -new audio effects -support for webM/ vp8 video playback and AAC audio encoding
2.3.3 Gingerbread	10	Feb 9 2011 -several improvements and API fixes -support for voice or video chat using Google Talk
3.0 Honeycomb	11	Feb 22 2011 -optimised tablet support -simplified multitasking - quick jump from one application to another -support for multi core processors

Versions	APK/API level	Release date and Activity
3.1 Honeycomb	12	May 10, 2011 -connectivity for USB accessories -resizable home screen widgets -support for FLAC audio playback
3.2 Honeycomb	13	July 15, 2011 -to access files on the SD card
4.0 Ice Cream Sandwich	14	October 18, 2011 -integrated screenshot capture -improved error correction -Face unlock feature -Android beam – a near-field communication feature -support for WebP image format -Wi-Fi direct - Android VPN framework (AVF) and TUN - QVGA resolution -smooth screen rotation -improved phone number recognition

Versions	APK/API level	Release date and Activity
4.1 Jelly Bean	16	June 27 2012 -triple buffering in the graphics pipeline -enhanced accessibility -turn-off notifications on application specific basis - USB audio (for external sound DACs) - audio chaining (gapless playback)
4.2 Jelly Bean	17	November 13 2012 -lockscreen improvements -daydream screensavers -SELinux support -premium SMS confirmation - group message support
4.3 Jelly Bean	18	July 24 2013 -Bluetooth low energy support -Bluetooth audio/video remote control profile (AVRCP) 1.3 support - system level support for geofencing and Wi-Fi scanning APIs - background Wi-Fi location still runs even when Wi-Fi is turned off - native emoji support

Versions	APK /API level	Release date and Activity
4.4 KitKat	19	Oct 31 2013 -applications can use immersive mode to keep the navigation and status bars hidden during user interaction -wireless printing capability -native infrared blaster API - enforcing SELinux
4.4 w KitKat with variable extensions	20	June 25 2014 -Android wear platform for smart watches -offline music playback -GPS support
5.0 lollipop	21	Nov 12 2014 -refresh lock screen -project volta, for battery life improvements -audio input and output through USB device -third party apps can read data from external storage like SD cards -recently used apps are remembered -addition of 15 new language like Kannada, Tamil ,Telugu, Chinese Sinhala - updated emoji

Versions	APK/API level	Release date and Activity
5.1 lollipop	22	March 9 2015 -Wi-Fi and to control paired bluetooth devices from quick settings -support for multiple SIM cards -device protection →remain locked when stolen until the owner signs in to their Google account -Native Wi-Fi calling support
6.0 Marsh mallow	23	Oct 5 2015 -introduction of Doze mode , which reduces CPU speed while the screen is off in order to save battery life -USB-c support -MIDI support for musical instruments (Musical Instruments Digital Interface)
7.0 Nougat	24	August 22 2016 -Unicode 9.0 – a computing industry standard for encoding, representing and handling of text, -emoji →ideograms /smileys or pictograms,/emoticons -daydream virtual reality platform (VR i/f) for use with a head mount for software - picture-in-picture support for Android TV
7.1	25	Oct 4, 2016 - Fingerprint sensor gesture to open/close Notifications - CDMA voice privacy property →mobile start encrypting the calls

Versions	APK/ API level	Release date and Activity
8.0 Oreo	26	Aug 21, 2017 -project Treble →easier for Hardware makers to deliver Android updates -Sony LDAC Codec support →an audio coding techniques for streaming audio over Bluetooth (wireless and high resolution)
8.1.0 Oreo	27	Dec 5, 2017 -neural network API -shared memory API - Android Oreo Go edition for devices with less than 1GB RAM
9 Pie	28	Aug 6, 2018 - DNS over TLS →security protocol to the system queries and answers on the transport layer security protocol for improve user privacy and security and dropping appointments support 3D graphics and computing API
10 (Queen cake or Quince Tart) unofficial	29	September 3 2019 -to access location in the background -AVI Video codec (AO media video 1) for video transmission over the internet -A native MIDI API -Better support for biometric authentication in apps -support for WPA3 (Wi-Fi protected Access for wireless computer

Versions	APK/API level	Release date and Activity
11	30	**Sep 8, 2020** -Chat bubbles. -Screen recorder. -Notification history. -New permissions controls. -API distinction between standalone 5G NR and non-standalone 5G. -One-time permission -Permissions auto-reset. -Wireless [Android Auto](#) on devices with 5GHz Wi-Fi. -Increased number of updatable core OS components in Google Play from 6 to 12. -Enterprise work profile privacy protections now apply on company-owned devices.

-**Bubbles** are Android's take on the Facebook Messenger Chat Heads interface. When you receive a **message** from Facebook Messenger, it appears on your screen as a floating **bubble** that you can move around, tap to view, and either leave it on your screen or drag it down to the bottom of the display to close it.

SPEECH BUBBLE ICON SET

Features of Android

- Open Source
  - Android is an open source operating system – the source code is open to public – dissimilar to iOS, which is kept secret by Apple.

- Storage
  - SQLite , a light weight relational database is used for data storage purposes

- Media support
  - Audio/video/still media formats supported are : WebM, H.263, H.264, AAC, HE-AAC(in 3GP or MP4 container), MP3, MIDI, WAV, JPEG, PNG, GIF, Ogg Vorbis, FLAC, BMP, WebP

- WebM – It is an open, high-quality video standard. It stores video compressed using VP8 technology and audio compressed using Ogg Vorbis compression. **WEBM** files are commonly **used** for delivering online videos using the HTML5 <video> tag.
- H.263 – Video compression standard, designed for video conferencing
- H.264 – Video compression technology or codec (better than DVD)
- AAC – Advanced Audio Coding is an [audio coding standard](#) for [lossy](#) [digital audio](#) [compression](#)
- HE-AAC – High Efficiency Advanced Audio Coding

- 3GP – a multimedia file saved in audio and video container format , developed by 3rd generation partnership project
- There are two kinds of compression: [lossless and lossy](). In order to compress a video, your file must also have a corresponding [codec](). A codec is a software that compresses your video so it can be stored and played back. The most common codec includes h.264, which is often used for high-definition digital video and distribution of video content. [bit rate]() refers to the amount of data stored for each second of media that is played. The higher the bit rate, the less compression, which results in overall higher quality. Also higher the bit rate, the larger the file size

– In addition to a codec, each video file has a [container](). The container is like a box that contains video, audio, and metadata (vital data such as captions, SEO, and vital information that pieces the video together for playback). It can also be called a file extension since they are often seen as file names, such as AVI, MOV, or MP4. MP4 h.264 is currently the industry standard for high definition video and provides the most practical way of distributing video content.

- MP4 by MPEG – a multimedia container format to store audio visual data
- MPEG4 SP – simple profile for low processing power coding
- AMR – adaptive multirate audio codec – for speech coding
- AMR- NB - narrow band for GSM
- AMR-WB - Wide band for UMTS (Universal Mobile Telecommunication System) for network based on GSM(Uses WCDMA technique)
- FLAC – Free Lossless Audio Codec
- MIDI
- WAV- Waveform Audio **File Format** (WAVE, or **WAV,** pronounced "wave") is an audio **file format** standard, developed by IBM and Microsoft, for storing an audio bitstream on PCs. ... It is the main **format** used on Microsoft Windows systems for raw and typically uncompressed audio.
- WebP - **WebP** is a modern image format that provides superior lossless and lossy compression for images on the web. Using **WebP,** smaller, richer images can be created that make the web faster. ... **WebP** lossy images are 25-34% smaller than comparable JPEG images

- Streaming media support
- Multitouch
- Web browserr
- Video calling
- Multitasking
- Accessibility
- Voice based features
- External storage

Android Architecture

- Android OS is a software stack of different layers, where every layer is a group of several programs components.
- It includes operating system, middle ware and important applications.
- Every layer in the architecture provides different services

- Android has the following layers :
  - Applications
  - Application Framework
  - Libraries
  - Android Runtime
  - Linux Kernel

Android Architecture
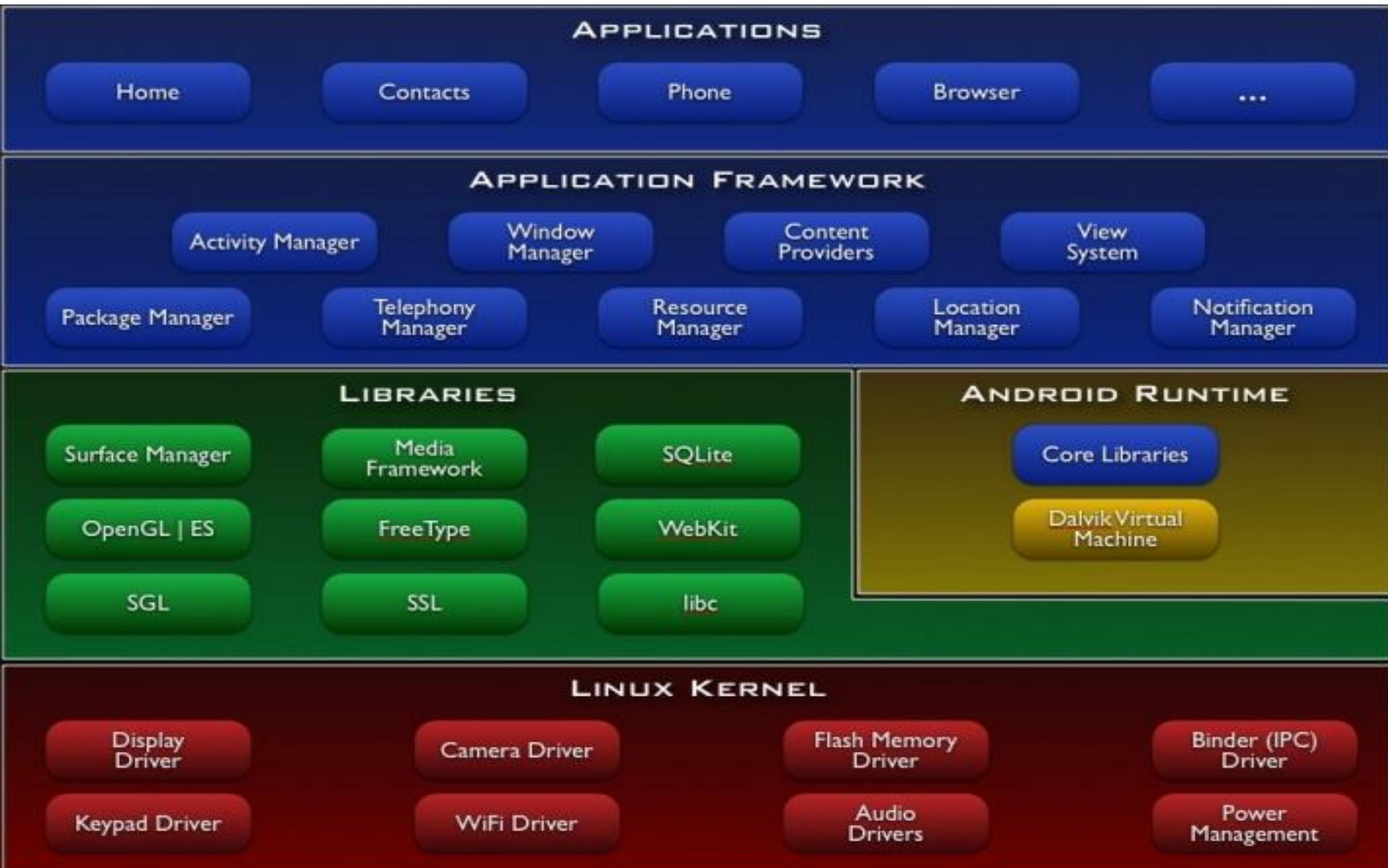
1. What is Android Architecture?

- Android is a mobile operating system that has an open-source framework and is based on **Linux** which helps us to develop advanced and user-friendly applications.

- Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.

- Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

- The main components of android architecture are the following:-
  - Applications
  - Application Framework
  - Android Runtime
  - Platform Libraries
  - Linux Kernel

  (here applications and framework are written in Java, which communicate through c & c++ core libraries through JNI-java native interface and linux kernel includes drivers for hardware, networking, file sytem access and inter-process communication)

- Applications
  - All applications in the application layer are written using Java language. The various core applications that an android device provides include
    - Email client
    - SMS program
    - Maps
    - Browser
    - Calender
    - contacts

Android, by default, comes with a set of applications that make android devices usable from the offset.

- **Home:** The Homepage on Android consists of launcher icons for commonly used applications that the end-user may want quick access to. The apps can be started by clicking on the launchers of these apps. At the very top of the screen, there are widgets that show network, battery level, date, and time.

- **Contacts:** Android, by default, provides a means to store and retrieve contacts. Contact information is shared across other apps to enhance functionality.

- **Messages:** Android provides the capability to send and receive SMS messages.

- **Email:** Android comes with native support for email services. Setting up an Android device requires a Gmail account. Setting up Gmail activates other email-dependent components on Android devices. Some email dependent features include security and recovery mechanisms. Another email dependent feature is access to the Play Store, a marketplace for Android applications.

- **Browser:** Android comes with a default browser.

- **Notification Drawer:** Swiping down on the screen exposes the notification drawer. It provides application events that the user should be aware of. Above the notification are a set of shortcuts to some commonly used device settings that the users can toggle. These settings include on and off toggles for various hardware components such as Bluetooth and Wifi. Long pressing these events enables us to navigate to their configurations page.

- This layer is also referred to as user-level in contrast to the layers below that are mostly tuned for application development. Application developers create and customize the experiences for their apps on this layer. The layers below the application layer are not customized by application developers. They are considered part of the system layer. These layers are customized by device manufacturers, Google android teams, or third parties who want to use the Android source code for their product or research.

*guru99.com

- Application framework
  - Android offers developers the ability to build extremely rich and innovative applications.
  - developers can take benefit of the device hardware, access location information, background services, add notifications to the status bar & so on.
  - An application framework is a software library that provides a fundamental structure to support the development of applications for a specific environment – gives a skeletal support to build an application
  - Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.

- The features of the Android operating system are available to us through API's written in form of Java classes. And, Android developers use these high-level services to build applications.

- It also consists of an Android **Hardware Abstraction Layer (HAL)** that allows the Android Application framework to communicate with hardware-specific device drivers. It acts as an interface for hardware vendors to implement. An android application uses HAL APIs to get commands from different hardware devices.

- Designed to simplify the reuse of components – any application can publish its capabilities and any application can use them (reusability of code and design)

- Templates to reduce development effort

- The application framework consists of following key services:
  - **Activity Manager:** That manages the life cycle of applications and provides a common navigation backstack (testing and debugging methods).
  - **Content provider:** allows applications to access data from other applications (such as contacts) or to share their own data
  - **Resource Manager:** provides access to non-code resources, such as localized strings, graphics and layout files.
  - **Notification Manager:** that enables all applications to display custom alerts in the status bar .
  - **View System**: A rich and extensible set of views that can be used to build an application , including lists, buttons, grids, textboxes and embeddable web browser .

- Application framework layer contains 10 basic components :

1. Activity Manager : manages the life cycle of applications and provides a common navigation backstack [Activities for email – list, compose, read. When user presses back, new activity is finished and popped off the stack]

Each navigation event triggers activation and deactivation of many activities in respective applications.

Some examples of the way applications can react to these transitions include pausing activity in a game, stopping music playing during a phone call.

2. Window Manager : a system service, responsible for managing the ordered list of windows, like which windows are visible and how they are laid out on the screen.

Windows are the slots where we can view our app user interface.

The Window Manager helps in delivering a customized app experience. Apps can fill the complete screen for an immersive experience or share the screen with other apps. Android enables this by allowing multi-windows for each app.

3. View Manager : Every item in a user interface is a subclass of the android view class. Pre-built views – button, check box, progress bar etc., A view can be comprised of multiple other views (composite view) eg – Radio group – multiple radio button object

4. Contents manager / Provider : allows applications to access data from other applications (such as contacts) or to share their own data

5. Resource manager : Provides access to non-code embedded resources such as strings, color settings & UI layouts, icons, audio and video files, animations, text files etc.

6. Notifications manager : Allows applications to display alerts and notifications to the user, through visual, audio or vibration signals.

   Some examples of internal triggers are low-battery status events that trigger a notification to show low battery. Another example is user-specified events like an alarm. Some examples of external triggers include new messages or new wifi networks detected.

7. Package manager : an API that manages application install, uninstall and upgrade

8. Telephony manager : gives information about the telephony services on the device

9. Location manager : to access devices geographical location or to fire an application specified intent when the device enters a given geographical location

10. XMPP service manager : Extensible messaging and Presence Protocol – XML based client/server architecture that allows users to access network using other protocols (for chatting, ringtone services)

- Android Runtime
  - In android runtime, there are core libraries (to write Android applications using standard Java programming language.) and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.
  - DVM runs .dex files which are built from .class files at the time of compilation and provide higherr efficiency in low resource environments
  - The DVM allows memory management, security, isolation and threading support
  - The runtime translates code written by programmers into machine code that does computations and utilizes android framework components to deliver functionality. Android hosts multiple applications and system components that each run in their processes.

- Libraries
  - This layer is communicated through application framework layer . The communication happens through Java Native Interface (JNI) package.
  - This layer contains 9 components. They are :
    - Surface manager
    - Open GLIES
    - SQLite database
    - Media Framework
    - SGL (Scalable Graphics Libraries)
    - FreeType
    - SSL(Secured Socket Layer)
    - Webkit
    - LibC(C Libraries)

- Surface manager
  - The surface manager is responsible for ensuring the smooth rendering of application screens. It does this by composing 2D and 3D graphics for rendering. It further enables this by doing off-screen buffering.
- Open GL|ES: Open Graphics Library for embedded devices- support for high performance 2D and 3D graphics
- SQLite : open source RDBMS, SQLite is a C-language library that implements a small, fast SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers
- Media Framework: Supports playbacks and recording of various audio, video and picture formats.
- SGL (Scalable Graphics Libraries) : This is responsible for implementing low level graphics by using JNI
- Free Type: Font Rendering, Android comes with a preinstalled fast and flexible font engine. This makes it possible for application developers to style components of their application and deliver a rich experience that communicates the developer's intent.

- SSL(Secured Socket Layer) : Android also comes with an inbuilt security layer to enable secure communication between applications on Android and other devices such as servers, other mobile devices, routers etc.,

- Webkit : web browser support , This is an open-source Browser engine used as a basis to build browsers. The default Android browser before version 4.4 KitKat uses it for rendering web pages

- Lib C : The core of Android contains libraries written in C and C++, which are low-level languages meant for embedded use that help in maximizing performance.

- **Android Runtime**
  - Consists of Dalvik Virtual Machine and Core Libraries
    - **Dalvik Virtual Machine (DVM)**
      - A kind of JVM used in android devices to run apps and is optimized for low processing power and low memory environments
      - Makes use of Linux core features like memory management and multi-threading

    - Dalvik VM includes several features for performance optimization, verification, and monitoring, one of which is **Dalvik Executable (DEX)**.
      - Java source code is compiled by the Java compiler into .class files. Then the dx (dexer) tool, part of the Android SDK processes the .class files into a file format called DEX that contains Dalvik byte code. The dx tool eliminates all the redundant information that is present in the classes. In DEX all the classes of the application are packed into one file.

Linux Kernel

- The root component of the Android System is the Linux Kernel. It is the foundational piece that enables all of Android's functionality, and the entire android OS is built on top of Linux Kernel
- **Linux kernel** is responsible for device drivers, power management, memory management, device management and resource access.
- This layer has 10 major components that are drivers
- Android run time gives .dex files which call the suitable drivers like
  - Display driver
  - Camera driver
  - Bluetooth driver
  - Flash memory driver
  - Binder driver(IPC)
  - USB driver
  - Wifi driver
  - Keypad driver
  - Audio driver
  - Power management driver

Configuration of Android Environment

1. Operating System
2. Java JDK
3. Android SDK
4. Android Development Tools (ADT)
5. Android Virtual Devices (AVD)
6. Emulators
7. Dalvik Virtual Machine
8. Difference between Java virtual machine and Dalvik virtual machine
9. Steps to install and configure Eclipse and SDK

1. Operating System
   - Windows xp/vista/7/10
   - Mac OS X
   - Linux
2. Java JDK
   - As android programs are developed in the Java programming language, JDK (Java Development Kit) has to be installed, which includes Java SE and Java Runtime (JRE)
3. Android SDK

   - The **Android SDK** (Software Development Kit) **is** a set of development tools that **are** used to develop applications for the **Android** platform

   - The **Android SDK** consists of an emulator, development tools, sample projects with source code, and the required libraries to build **Android** applications
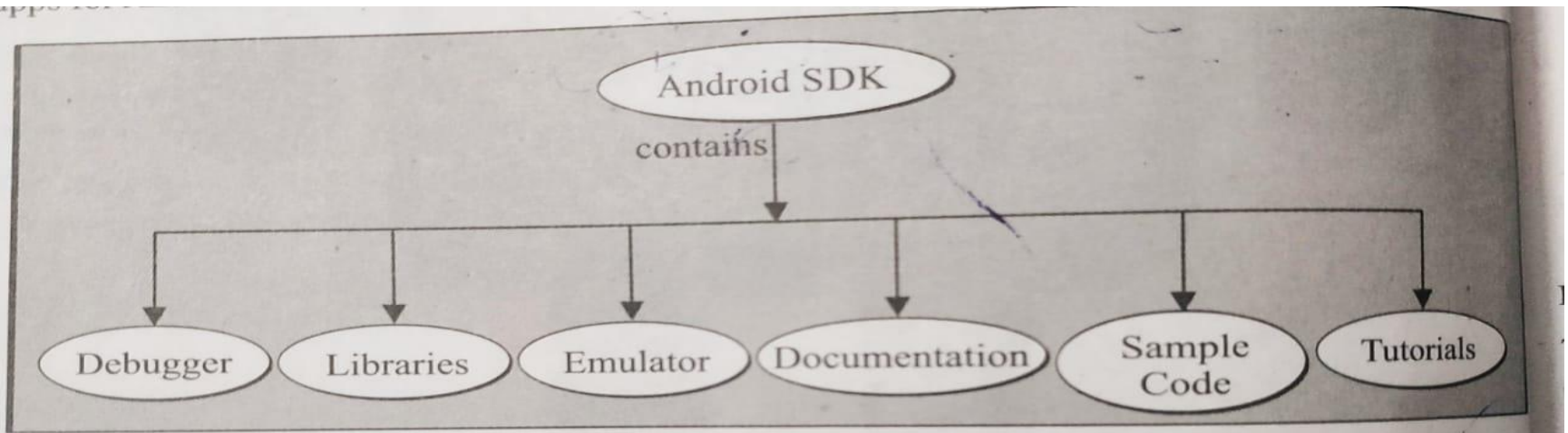
Figure 2.1: Android SDK

Android SDK can be downloaded from http://developer .android.com/sdk/index.html using Android SDK manager

*Android SDK is made up of two main parts : the tools and the packages
Base tools are executables and supporting file that helps application development
Packages are records specific to a particular version of android(called a platform) or a particular add-on to a platform
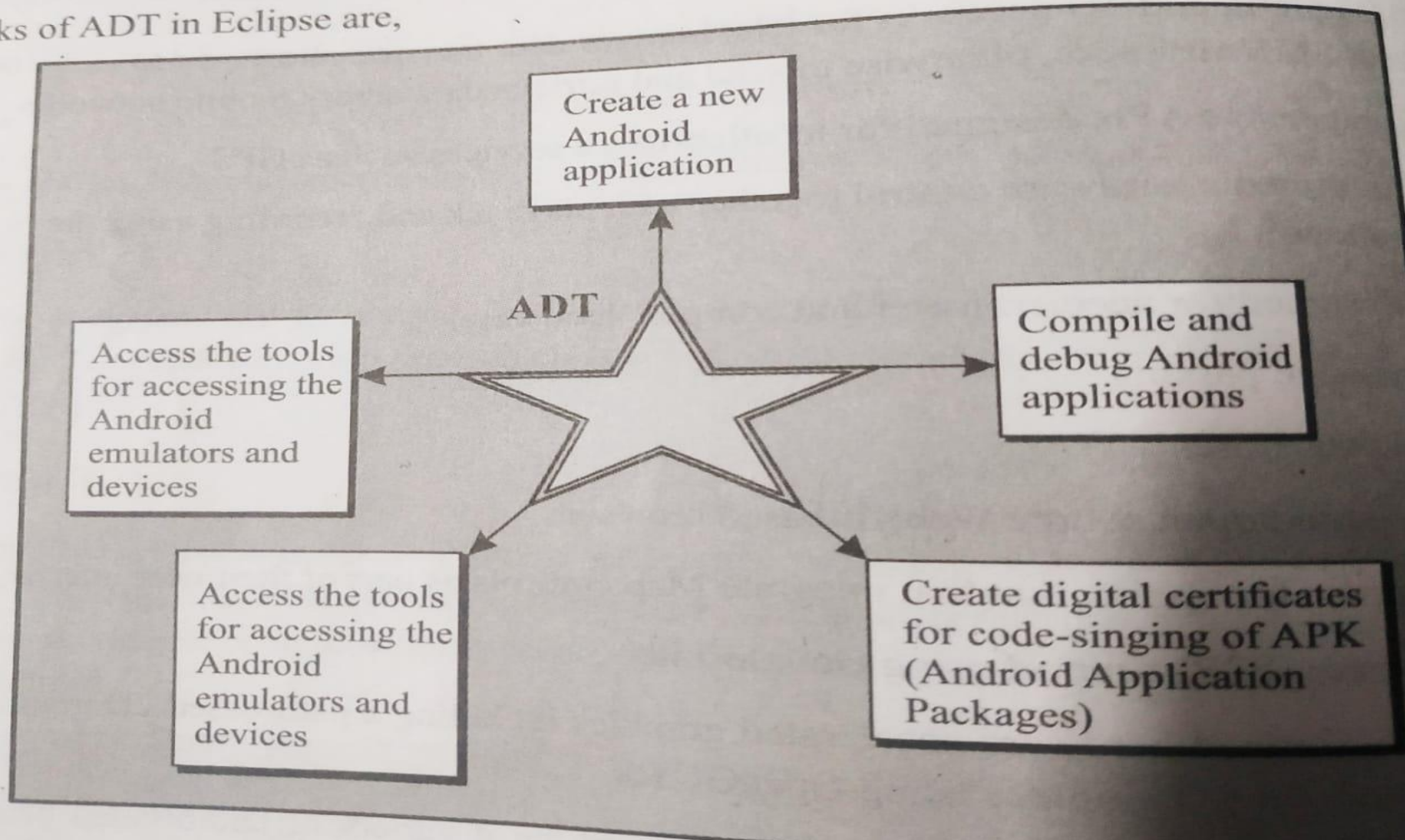
- Some Android SDK features include :
  - No licensing , distribution or development fees (open source)
  - Wifi , hardware access
  - GSM, EDGE (Enhanced Data rates for Global Evolution – mainly used for data in cell phones) and 3G networks for telephony or data transfer – allows us to make calls or SMS messages
  - Comprehensive APIs designed for location based services such as GPS
  - Full multimedia hardware control with playback and recording using the camera and microphone
  - APIs designed for accelerometer and compass hardware
  - IPC message passing [Process can communicate with (i) shared memory and (ii) message passing]
  - Shared data stores
  - An integrated open source WebKit (a browser engine developed by Apple) based browser
  - Support for application that integrate Map controls as part of their user interface
  - Peer-to-peer (p2p) support using Google Talk (**Google Talk** was an instant messaging service that provided both text and voice communication.)
  - Mobile-optimized hardware-accelerated graphics includeing a path-based 2D graphics library and support for 3D graphics using Open GL ES
  - Media Libraries designed for playing and recording a variety of audio/video or still image formats
  - An application framework that encourage reuse of application components and the replacement of native applications
  - Platform used – Eclipse Galileo(V3.5 or higher), Eclipse Helios(V3.6.2 or higher), Eclipse Indigo

4. Android Development Tools(ADT)

- - ADT is a plugin for the Eclipse IDE which provides a suitable environment to develop an Android application where creation, compilation and debugging are possible . ADT can set up new Android projects, create an application UI, insert packages based on the Android Framework API, debug your application using the Android SDK tools and export .apk files in order to distribute the application

- Also ADT provides custom XML editors and debug output pane

- Android Developer Tools Bundle (ADT Bundle) provides,
  - Eclipse + ADT plugin
  - Android SDK tools
  - Android platform tools
  - The most recent Android platform
  - The newest android system image for the emulator

ppncation.

rks of ADT in Eclipse are,

Create a new
Android
application

ADT

Access the tools
for accessing the
Android
emulators and
devices

Compile and
debug Android
applications

Access the tools
for accessing the
Android
emulators and
devices

Create digital certificates
for code-singing of APK
(Android Application
Packages)

5.   Android Virtual Devices (AVD)

   - An Android virtual device AVD is an emulator configuration that enables to model an actual device by calling hardware and software options to be emulated by the Android Emulator

   - the AVD manager can be used to create an AVD. launch it from eclipse by clicking Window| AVD manager

   - an Android virtual device AVD is used for testing the Android applications. An AVD is an emulator occurrence that enables to form a real device Each AVD consists of a hardware sketch, a connection to a system image , and emulated storage , such as a secure digital SD card

 An AVD consists of

> A hardware profile: defines the hardware features of the virtual device. for example, you can describe whether the device has a camera, whether it uses a physical QWERTY keyboard or a dialling pad , how much memory has and so on.

> A mapping to a system image: what version of the Android platform will run on the virtual machine can be defined. The version of the standard Android platform or the system image packaged with the SDK add on can be chosen

> the emulator skin to be used with the AVD can be specified , which lets you control the screen dimensions, look and so on .The emulated SD card to use with the AVD, also can be specified

> A dedicated storage area on your development machine: the devices user area (installed applications settings and so on) and emulated SD card are stored in this area

•

- Difference between JVM(Java Virtual Machine) and DVM(Dalvik Virtual Machine)

DVM (Dalvik Virtual Machine)	JVM(Java Virtual Machine)
1 It is Register based which is designed to run on low memory.	It is Stack based.
2 DVM uses its own byte code and runs the ".Dex" file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs ".class" file having JIT (Just In Time).
3 DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance.	A single instance of JVM is shared with multiple applications.
4 DVM supports the Android operating system only.	JVM supports multiple operating systems.
5For DVM very few Re-tools are available	For JVM many Re-tools are available.
6There is a constant pool for every application	It has a constant pool for every class.