mobile Computing sec 1

→ what is android?

- a soft ware platform and operating system for mobile devices.
- Based on:-the Linux Kernel.
- Developed by:-Google and open Handest Alliance (OHA).
- Allow writting managed Code in the Java language.

→ History of Android

2005 → google buys android Inc.
     → Work on Daluilk VM starts.

2007 → OHA Announced.
     → Early look SDK

2008 → T- Mobile G1 Announced.
     → SDK 1-0 Released
     → Andriod open Sourced.

→ Open Handest Alliance:-

- group of 47 tech. and mobile Companies have Come together to accelerate innovation in mobile and offer Consumers a richer, less expensive, and better mobile experience.
     → developed Andvoid ™, the first, Compelete, open, free mobile platform.

→ **platform:-** Hardware

- android isn't single piece of hardware it's a compelete end-to-end software ⟶ that can be adapted to work on any number of hardware configurations.

→ **Stackholders:-**

- network operators want to lock down their networks, controlling and metering traffic

- Device manufactures want to differentiate themselves with features, reliability, and price points.

- Software vendors want complete access to the hardware to deliver cutting-edge applications.

→ **app. framework**

accesses these core libraries through the Dalvik VM, the gateway to the android platform.

→ **Some features :-**

- **view system :-** used to build an app., including lists, grids, textboxes, buttons, and embedded web browser.

- **Content provider :-** Enabling apps to access data from other apps or to share their own data.

- **Resource Manager :-** providing access to non-code resources ex. (localized string, graphics, layout files)

- **Notification Manager:-** Enabling all apps to display customer alerts in the status bar.

– **Activity Manager :-** Managing the lifecycle of apps and providing a common navigation back stack.

→ **Android S/W stack - Libraries**

– Including a set of (C/C++) used by components of the Android system.

– Exposed to developers through the Android app Framework

– based on paket video's / open CORE

– responsible for recording and playback of audio and video

→ **Surface Manager :-**
   – access to the display system and supports 2D, 3D

⇒ **web kit ⟶**

   – responsible for browser support, it is the library that support ⟶ Google Chrome, Apple Inc's Safari.

→ **Free Type: -**

– responsible for font support.

→ **SQ Lite: -**
   – is relational db that is available on the device itself
   – an independent open source effort for relational db and not directly tied to Android.

→ **Core Libraries**

– providing most of the functionality available in the core libraries of the Java languuge

— API_s
- Data structures - Utilities - File Access -
- Network Access - Etc.

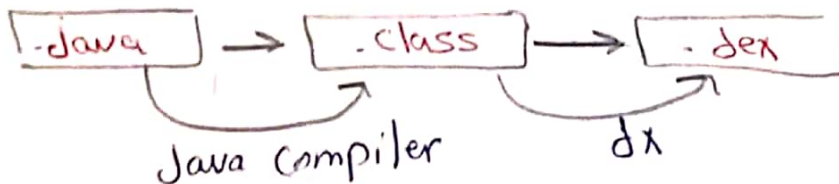⟶ the Dalvik Virtual Machine

- is optimised for mobile app ⟶

1- Run multiple VMs efficiently.
2- Each app has its own VM.
3- Minimal memory footprint.

→ Android S/W stack - Runtime

→ Dalvik VM

⤷ executing the Dalvik executable (.dex) format.
  --dex format is optimized for minimal memory footprint
  - Compilation

| .java | → | .class | ⟶ | .dex |

Java compiler   dx

⤷ Relying on (the linux kernel) for:-

→ threading
→ Low-level memory management

→ Linux Kernel

→ Relying on linux kernel 2.6 for core system services
- memory and process Management.
- Network stack.
- Driver Model
- Security.

→ the supplied device drivers include Display, Camera, Keypad, wifi, flash memory, Audio, and IPC

(interprocess communication)

وظيفته
→ providing an abstraction layer between the H.w and the rest of the S.w stack.

→ **platform**
   → supports wireless communication using
      - GSM mobile-phone ~~the~~ tech.
      - 3G.
      - Edge
      - 802.11 wifi networks.

→ **Sw Development.**
   - Java
   - Android SDK } Requirements

- Android SDK. (tools)
   -class library.
   - Developer Tools.
   - Emulator and system Images
   -Documentation and sample Code.

} - Eclipse IDE + ADT
   +Reduce Development and testing time.
   - Makes User Interface - Creation easier
   - Makes app Description easier

→ **Adv.**
   - available for windows, Mac, linux.
   - built on java.
   - you can release your own app on your own blog.

→ **App Building Blocks.**
- Activity.
- Intent Receiver.
- Service.
- Content provider.

→ Activity
- typically correspond to one UI screen.
- But they can:
  - ↳ Be faceless.
  - → Be in a floating window.
  - → Return a value.

→ Intents

2 Components مع الاشارة (الى)

- a verb and object; a description of what you want done
- activity and Intent Receiver describe what Intents they can service.

---

2 → Intent Receiver.
- Components that respond to broadcast 'Intents'
- way to respond to external notification or alarms.
- Apps can invent and broadcast their own Intent.

---

3 → Service
- faceless Components that run in the background.
  ex. music player, network download, etc...

---

4 → Content providers
- enables sharing of data across apps → ex. address book. photos
- Provides uniform APIs for: → Querying, delete, update, insert.
- is represented by [URI] and [MIME] type