

## Syllabus

# Android Lecture 01

Development of Android Applications :

UNIT- (1)

(1) Introduction to ANDROID

1. Android
2. DVM
3. .apk file Extension
4. fundamental
5. Basic Building Block

Activities

Services

Broadcast Receivers

Content Providers

6. UI COMPONENTS

Views

Notification

7. Components for Communication

Intent

Intent filters

8. Android API Levels.

## Unit -②

### ② Environment Setup & Basic Project Structure

• ① Setting up development Environment Android

2. Manifest. xml

3. gradle

4. Uses-permission & uses sdk

5. Resources

6. R. java

7. Assets

8. Layout

9. Drawable Resources

10. First Sample Application

11. Launching Emulator

12. Editing emulator Settings

13. Emulator Shortcut

14. logcat Usage

15. DDMS

16. Hello World APP

Debugging Android App.

### UNIT-3.

## ③ ANDROID FUNDAMENTAL & UI Design :

Activities

Activity Lifecycle

Permission System.

Basic UI Components

TextView

EditText

Button

Radio Button

Image View

Check Box

Progress bar

Event handling in Android.

layouts

Linear Layout

Relative Layout

frame Layout

Coordinate Layout { dip  
dp  
spip  
sp  
{ px

Intent:  
introduction

Types ( Explicit  
Implicit )

## Unit - ④

### ④. Menus & Preferences :

introduction

Types { option  
Context

uses of Shared preferences.

## Unit - ⑤

### ⑤ Advanced UI Components

Time & Date

List View

Grid View

Card View

recycler View

- Popup
- fragments
- material design.
- floating button
- Toof bar.

Adapters { Base Adapter  
Array Adapter

View holder

Dialogs

Toast

Unit-⑥

## 6. Threads in Android

1. Threads running on UI thread
2. Worker thread
3. Handler & Runnable
4. AsyncTask
5. Calling Web Services
6. Consuming JSON data from Web Services

Unit-⑦

## 7. Notification & Services

1. Broadcast Receivers
2. services / Types
3. implementing a service
4. Service lifecycle.

## Unit - 8

### 8. Storage & Content Provider

#### 1. Supported Storage in Android

{ Internal memory  
External memory  
Shared preferences & n/w

#### 2. SQLite introduction

#### 3. CRUD operations in SQLite database

{ Cursor  
Content Values

#### 4. Basics of Content provider.

## List of Practicals :

1. Install the Android studio & Setup Development environment.
2. W.A.P. to demonstrate activity  
→ Activity Life Cycle.
3. W.A.P. to demonstrate different types of layouts.
4. W.A.P. to implement Simple Calculator using text view, edit view, option button & button.
5. W.A.P. to develop app having multiple activities & User should be able to switch b/w the activities by using intents.
6. W.A.P. to demonstrate List View

7. W.A.P. to demonstrate photo gallery
8. W.A.P. to demonstrate Date & time picker
9. Dev. an app with Context menu & Option menu.
10. W.A.P. to demonstrate f.<sup>n</sup> of Shared Preferences.
11. Dev. an app similar to gmail application.
12. W.A.P. to demonstrate Service
13. W.A.P. to demonstrate application of intent class
14. W.A.P. to create a text file in external memory.
15. W.A.P. to store & fetch data from SQLite database.

# Unit ① INTRODUCTION TO ANDROID

Android ?

It is open Source  
Linux-based OS. for mobile devices & tablets.

developed by - OHA (Open Handset Alliance)  
led by Google.

first version of Android SDK released in 2007

Android 1.0 → 2008

Android 4.1 → 2012 [Jelly bean]

every updated Android version comes with  
better UI, better functionality, better Performance  
mobile application development.

- \* mobile application development.
- \* zero development cost.
- \* open Source
- \* multi platform Support.
- \* multi carrier Support.
- \* open distribution model.

## features of Android

Rich UI - provide intuitive user Interface

Connectivity - GSM | EDGE, IDEN, CDMA, UMTS  
Bluetooth      WIFI      LTE      INIMAX etc.

Storage - SQLite: a Lightweight relational DB

media Support      MPEG      AMR      AAC      MP3      MIDI  
                        WAV      JPEG      PNG      GIF      BMP

messaging - sms      mms

Web browser - based on open Source Webkit layout  
                        engine & Javascript engine  
                        supporting HTML5      CSS3

multi touch

multi tasking -

resizable widgets

multi language .

WIFI-Direct .

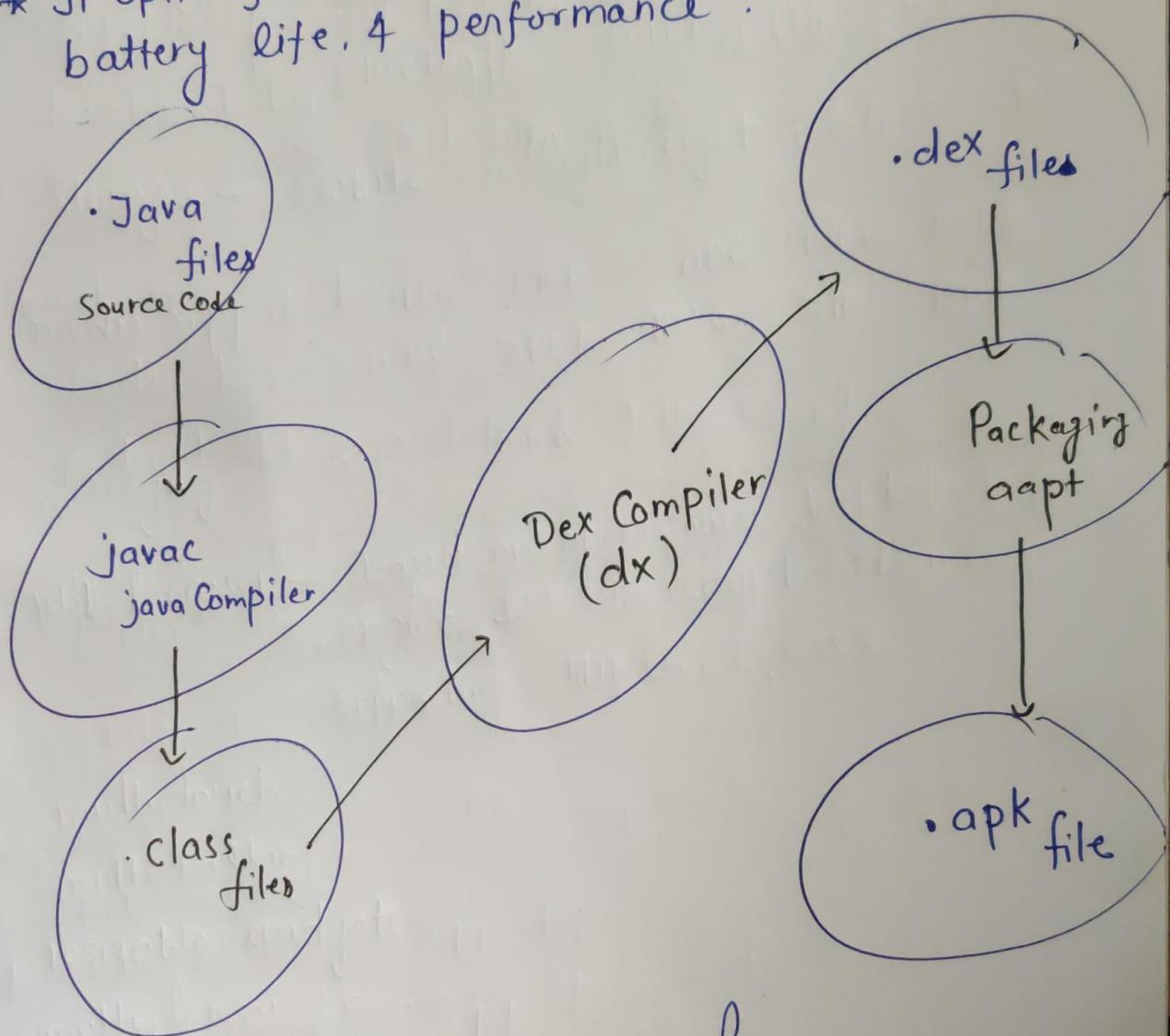
Etc.

Android Beam - NFC service ( instant share )

# DUM ( Dalvik Virtual Machine )

developed by - Dan Bornstein

- \* It optimizes the virtual machine for memory, battery life, & performance.



Compiling & packaging process.

- \* multiple class files are converted into one dex file.

javac tool → compiler java source file into class file.  
dx tool → compiler all class files into single dex file.

aapt tool → Android assets Packaging tool  
handles packaging process.

\* executable file is apk

.apk → it is executable file  
full form → "Android Application Package"  
used to distribute application's on android OS  
based devices.

## Basic Building blocks :

- \* Application components are essential.
- \* Components are coupled in AndroidManifest.xml file  
this file describes each component of application & how they interact.

4 main components of Android Application:

1. Activities — (UI)
2. Services — (background processing)
3. Broadcast Receivers — (communication b/w android OS and application)
4. Content providers — (handle data 4 DBMS issues)

# Android Studio S/w Explataion

XML.

AndroidMenifest.xml.

Activity.

Gradle.

Android SDK.

AUD | Emulator.

How to configure mobile with Android studio.

open Android studio >> new Project >> Empty View Activity

create new Project

>> Now your Project will download gradle files

⇒ \*\* while developing a project you should be connected to internet.

→ Project Structure

Editor

Device Manager

APP

manifest  
java  
res

→ AndroidMenifest.xml

by default  
file

gradle scripts

XML -

- \* Extensible markup language
- \* front end (UI design)
- \* tag based language

<tag> attribute "value"

X AndroidManifest.xml

→ It provides details of application

1. How many activities we have created and their names.
2. which activity should open first.
3. App theme, icons, permissions

<?xml >

<manifest xmlns:android = "url"  
package = "com.example.Project" >

<uses-permission android:name = "" >

<application >

    android:allowBackup = "true"

    android:icon = "@mipmap/ic\_launcher"

    android:label = "apk\_file\_name"

    android:supportRtl = "true"

    android:theme = "@style/Theme.Project"

    <activity >

        android:name = "MainActivity"

        android:exported = "true"

        <intent-filter >

            <action >

            <category >

            </intent-filter >

            </activity >

    </application >

  </manifest >

→ { can  
contain  
multiple  
activity.  
with  
preference.

Activity :-

Android screen is known as an Activity.



Activity = .xml + .java

.xml (UI front end) ↗ under res folder

.java (backend - action) ↗ Under java folder.

package right click  
new >> activity.

java - MainActivity.java

import androidx.\*;

import android.\*;

public class MainActivity extends AppCompatActivity

{

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

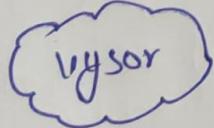
## Combination of scripts.

gradle :

- \* qt is an open source build system which is used to automate building, testing, deployment etc. of our application.
  - \* qt is used to generate the .apk file from .java & .xml file.
- Gradle (build.gradle) contain the information about min sdk version, target sdk, android app version.

SDK

Emulator

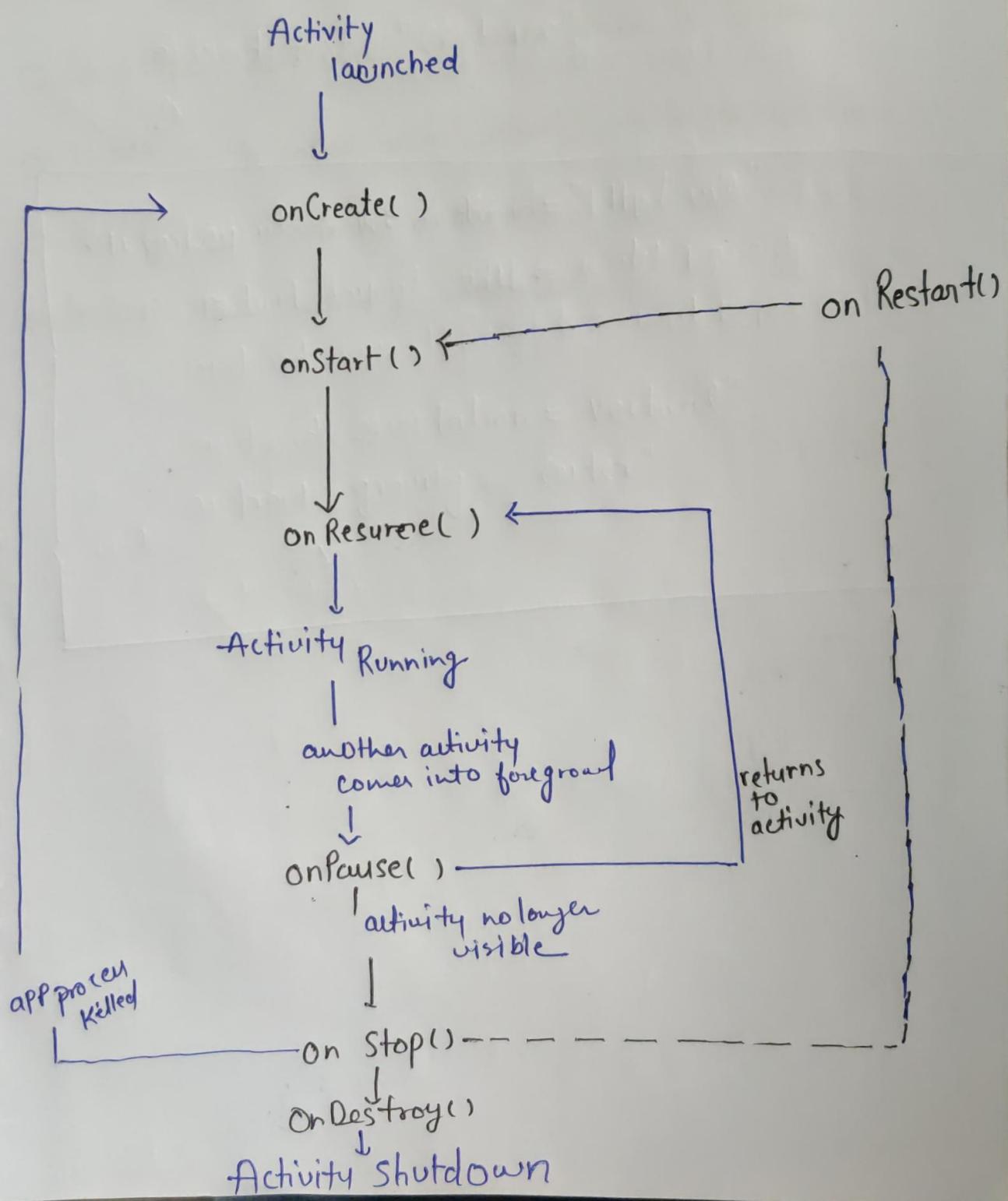


```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Click me"  
/>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:gravity="center"  
/>
```

# ANDROID ACTIVITY LIFECYCLE

starting call method - `onCreate()`



`OnCreate()` — activity is first created, first callback.

`OnStart()` — is called when activity visible to user.

`OnResume()` — is called when User  $\xrightarrow{\text{interacts}}$  with Application.

`OnPause()` — can't execute any code when it is called.

`OnStop()` — is called when activity no longer visible.

`onDestroy()` — activity destroyed.

`onRestart()` — activity restarted.

## MainActivity.java

```
package pkg_name;  
com.example.Project01;  
  
import android.*;  
    . os.Bundle;  
    . app.Activity;  
    . util.Log;
```

```
public class MainActivity extends Activity
```

```
{
```

Activity class loads  
all UI component  
using XML file  
res/layout/activity\_main.xml

```
String msg = "Android:";
```

```
public void onCreate(Bundle savedInstanceState)
```

```
{
```

```
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);
```

```
Log.d(msg, "The onCreate() event");
```

```
3
```

```
public void onStart()
```

```
{ super.onStart();
```

```
Log.d(msg, "The onStart() event");
```

```
3
```

```
: 3
```

⇒ an application can have one / more activities

activities are defined under → *AndroidManifest.xml*  
main activity declared with → <intent-filter>

### - *AndroidManifest.xml*

```
<?xml  
<manifest>  
<application  
    android:  
        <activity android:name = "MainActivity">  
            <intent-filter>  
                <action android:name = "android.intent.action  
                    .MAIN" />  
                <category android:name = "android.intent.  
                    category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

if action &  
category are  
not declared  
then your  
app icon  
will not  
appear in  
Home  
Screen  
list of  
apps.

Practical 01 : *How to install & configure Android studio*

Aim / OBJECT :

System Requirements :

OS : Windows 10 / 11 / Mac Os / Linux (64bit)

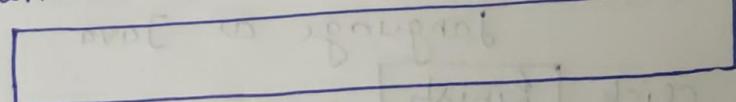
Java { JDK - 21 or higher  
JRE - 1.8 or higher

IDE - Android Studio  
RAM → 8 GB or higher  
HD → 50 GB or higher

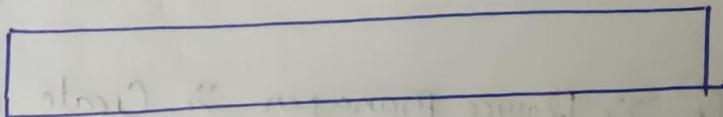
Procedure

Step 1 : download Android Studio

link



Step 2 : run the download installer file



Step 3 : Installation

launch androidstudio.exe.

→ \* before installation machine must have JDK & JRE installed on machine.

→ Click next

→ Now mention JDK path in Android Studio installer.

→ Choose components { Android Studio ✓  
SDK ✓  
AVM ✓

→ specify focal machine path

→ specify Ram space - 2GB (recommended)

→ Now extract SDK packages into our local machine.

→ select form factors min SDK (download it via SDK manager).

File > new Project

Select empty views Activity

→ specify Project name

package name

min SDK level

language as Java

click **finish**

→ Tools > Device Manager > Create new AVD  
(virtual device)

or

Connect Physical device through  
usb debugging enabled

Code X

SW → VYSON

Result: installation successful.