Date : / /
Page :

Android overview -

source code - open source.

Kernel - linux (full security).

OHA → Google (2007 - SDK launch).

Alliance.

2008. sept → first official version of SDK is released

Android 1.0.

71% → market share Android

21% → IOS capture market share

Features -

→ Cost effective

→ open source

→ security

→ connectivity

→ Media support / Image support

→ open GLES (C++) for 2D & 3D graphics

→ storage

1-tier application

contacts, Notes, alarm, etc.

↓

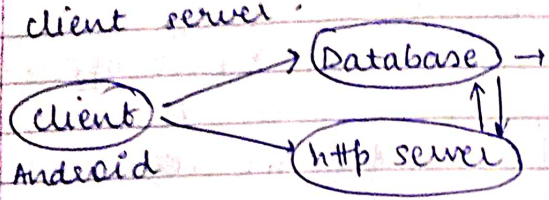
local database (mobile space).

mango db → connectivity
Android studio.

Date: / /
Page:

2-tier application -
local storage + cloud / http.

3-tier application -
client server.



OLA, UBER, Amazon.

Android	IOS
open source	proprietry / close
71% share	21% share.
freedom to work on 64 bit or 32 bit not hardware compatible.	64 bit (sab mera hoga).
c/c++/java/kotlin (c/c++ → functionality; java → official language of android)	c/c++/obj-c/ swift.
XML file / java file.	

Date: / /
Page:

Android is an open source & linux based operating system for mobile devices such as smart phones, tablets, etc.

Android was developed by open handset Alliance (OHA) which was lead by google and other associated companies.

The first beta version of android software was launched in the year 2007 as software developm kit (SDK).
The official version of android was released on september 2008 as android 1.0.

→ Features -

1) Android is open source OS. that means customisation, updation in the available code can be done as per requirement.

2) Android supports different types of connectivity like GSM, CDMA,

wifi, bluetooth for telephonic communication and data transfer. The advanced wifi technology enables the user to play games or to connect with other applications.

3) Android also contains multiple API's to support location tracker service

4) It provides file manager which can be helpful in managing data as well as storage.

5) It supports a wide range of media like AVI, MKV, MPEG4, etc. to play or record a variety of audio and video.

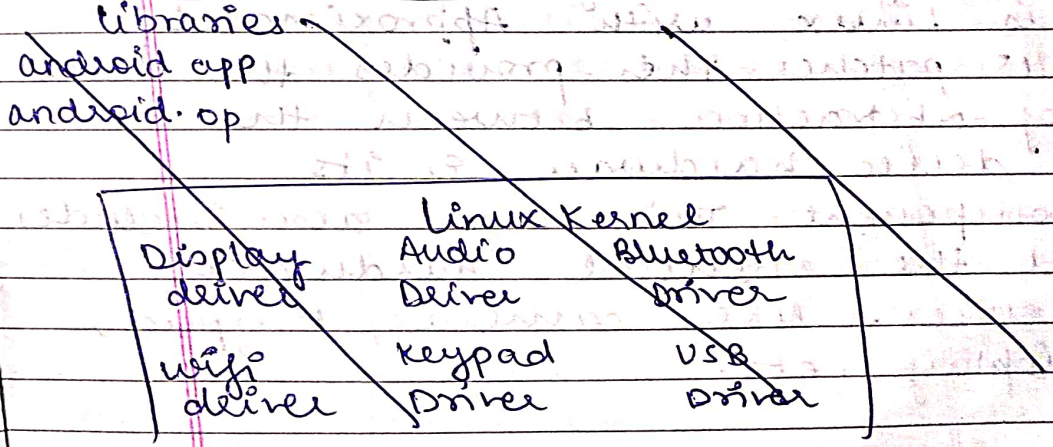
6) It also supports different image formats such as JPEG, PNG, GIF, etc.

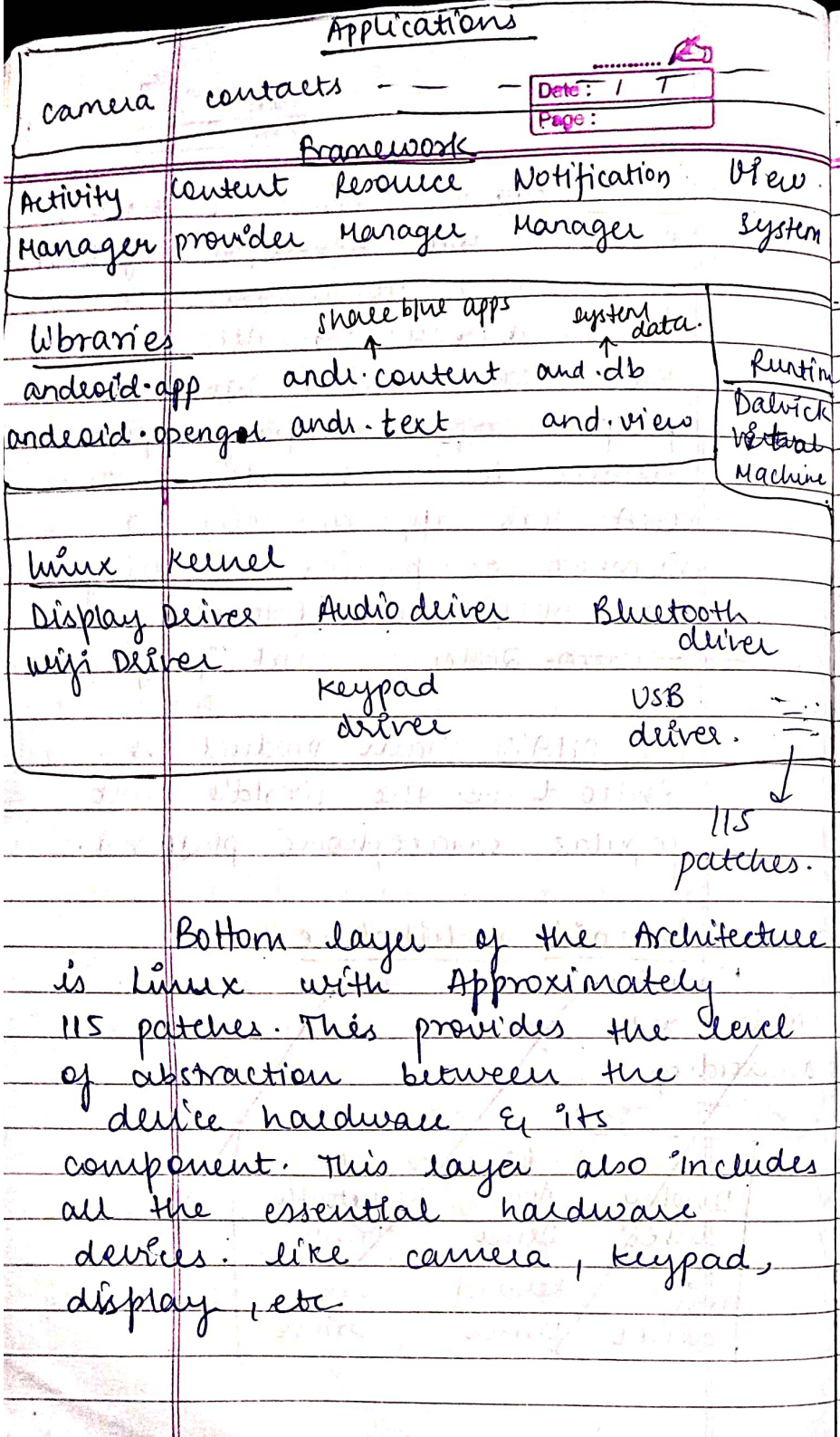
7) It facilitates virtual reality or 2D/3D graphics.

OHA - Open Handset Alliance.
Android was developed by OHA. or Open Handset Alliance i.e is a business Alliance that was created for the purpose of developing open mobile device standards. The OHA has approximately 80 member companies including HTC, DELL, INTEL, Motorola, ~~Telecom~~ Qulcom, and Google, etc.

The OHA's main product is Android i.e the world's most popular smartphone platform.

Android Architecture -





Bottom layer of the Architecture is Linux with Approximately 115 patches. This provides the level of abstraction between the device hardware & its component. This layer also includes all the essential hardware devices. like camera, keypad, display, etc.

- The second layer above the kernel layer is Android libraries. This layer consists of a set of Android libraries including open source web Browser SQ lite Database which is useful for storage & sharing of Data among Applications.

Some of the core Android libraries are -

1) ~~Andioid~~ andeoid.app
This Library provides an access to the applications model and is the cornerstone for all andeoid applications.

2) andeoid.content
This library facilitates, publishing and Messaging between different Application components.

3) andeoid.database
It is used to access the data that is published by the content provider. It also includes SQLite Database Management classes.

4. android.opengl –
provides the java interface to OpenGLES 3D Graphics rendering API's.

5. android.text –
used to render and manipulate the text on the device display.

6. android.view –
provides a fundamental building block of the application user interface.

Third section under this layer is android runtime. and it is available on the second layer. This section provides the key component called Dalvick virtual machine (DVM), kind of java virtual machine specially designed and optimised for android code.

The next layer application framework layer which provides many high level services to the application in the form of java classes. Some key services are:

1) Activity Manager –
controls all the aspects of the application lifecycle and activity stack.

2) content provider –
this allows the application to publish and share the content among multiple running applications.

3) Resource Manager –
It provides access to the non coding section, such as colour setting, strings, user Interface, etc.

4) Notification Manager –
this allows multiple applications to display alert and notification for the user.

5) View system -
In this an extensible set of views are used to create application user interface.

Last layer in the android architecture is the application layer which includes all the installed applications for example contacts, browser, games, etc.

AVD - Android Virtual Device - Emulator

Android Emulator is an Android Virtual Device which represents a specific Android Device.
Android Emulator can be used as a targeted device to execute and test your applications.
Android Emulator provides almost all the functionalities of a real device. We can also get the incoming phone calls, text messages, browse the web page, play music, or even download any application from the google playstore.
Testing android applications on Emulator are faster and easier than testing on a real device.
Android Emulator comes with free predefined configurations for several android phones, tablets, android tv, etc.
To run an android Emulator during your require :
1) SDK tool 26.1.1 or higher.
2) 64 bit processor.
3) Windows with CPU which supports unrestricted guests.
4) HAXM 6.2.1 version or higher.

Types of files -
Android studio provides numerous types of files with different utilities. Some of the files are enlisted below -

Date: / /
Page:

AndroidManifest.xml-
This file can be retrieved under the android route "app".
Under this file architecture android manifest folder and its xml file can be retrieved. This file facilitates the application with inbuilt pluggins, services on real time basis.

Activity Main.java-
Under android file hierarchy Activity Main.java file can be retrieved inside the java folder. This file provides the functionality of the UI components designed and developed in the layout file.

res folder-
① colors.xml
Under the res folder colors.xml

Date: / /
Page:

file can be retrieved. It includes the hexa code of the colors under the resource tag <resources>.
for eg- <resources>.
<color name = "xyz" > #FFFFFF
</color>
</resources>

drawable folder-
Under this folder all the media files, resources, audio, video, images, etc. are stored to include in the building android project.

layout folder-
① Activity_main.xml-
This file includes the user interface components such as Activity layout, buttons, textview, edit text, etc. This file is responsible for the UI part of your application. It builds the bridge b/w emulator and the UI frameworks.

② Mipmaps-
TO include the icons, image assets. this file is used.

You can generate or launch a icon, action button, notification icon, tab icons, etc.

Values Folder -

① string.xml -
Under the file heirarchy of android studio, strings.xml file can be retrieved under values folder. This file contains all the string resources and the values of different strings by a unique name. This file also stores string array by using xml language.

② R.java -
This is a resource type file which is autogenerated by android studio. It sums up a summary of the projects developed as the name of the class and the UI components as the attributes of the class. If deleted, android studio auto generates its again.

Updation and manipulation of the developer under this file not allowed. This file is generated and used by android studio itself.

→ Dalvick Virtual Machine (DVM) -
A modern JVM is high performance and provides excellent memory management. But it needs to be optimised for low powered handheld devices. Therefore, DVM is used. It is an android virtual machine which is optimised for mobile devices. It optimises the virtual machine for memory, battery life, better performance, network usage, etc.
The word Dalvick came from the city in Iceland and Dalvick Virtual Machine was written by Dan Bornstein.

Working of DVM -

- The java compiler that is java c converts the source file into

*.dex only the file which works on emulator, not any other file;

- java byte code i.e. .class.
- A DEX compiler converts this .class file into dalvick byte code. i.e .dex file.
- This conversion of multiple .class files are converted into a single .dex file.

Q2) Create an Android Application to change the background color on button click.

Q3) Create a simple Application in Android to change the colors of the background of a text box with the click of a button.

Q1). Create an Android Application to print a simple text on screen.

→ Advantages of DVM

- DVM supports the android operating system.
- In DVM the executable file is APK.
- Dalvick Virtual Machine supports .apk files to get the execution done.
- Under this the execution is faster.
- DVM is designed in such a way that a single device can run multiple instances of virtual machine very effectively.

→ Disadvantages of DVM -

- It only supports android OS.
- DVM requires more no. of instructio their registered machines to implement the same high level code.
- It consumes more time to convert files into DEX files.
- It requires more internal storage as compared to other compilers.

→ DDMS
(Dalvick Debug Monitor Service)
DDMS provides many services on the device which includes message formation, cell spoofing, capturing screenshot, exploring internal threads and file systems, etc.

DDMS can be run under android 3.0 versions. Under those versions click on the tools, then select android and then click on android device monitor. (ADM). Under this each Application runs in its own process and each process runs in the virtual machine (VM). Each VM exposes a unique port that a debugger can attach to. When a DDMS starts it connects to the ADB i.e. Android debug bridge.

that lets you communicate with an emulator or any connected android device.
DDMS helps in making sms to emulator by using telnet client and server.
DDMS is now depriciated and is replaced by Android Profiler. It helps in evaluating the profile of your apps CPU, your memory or network usage. In other words it provides real time information to help you understand how your app uses CPU, memory, network and battery resources. You can open the profiler window from the ribbon by clicking on tool tab and then selecting profiler option.

Date : / /
Page :

UNIT-2

App Component : Activity

lab. Steps - How to create file ourself.

1) Delete MainActivity.java.
2) Delete activity_main.java in res folder.
3) Right click on java → com.example - - - - options - new → java class → name - open.
4) res → layout → new → resource file → name it (don't use capital letters). → then fix layout as ~~co~~ linear. → enter.
5) In MainActivity1.java, we will write public class MainActivity1 extends AppCompatActivity {.
6) @override - jab isko call karein to yeh call ho by default.
7) public static void onCreate () {
--- R.content.Layout
- - - -
findViewById (R.id.B1);

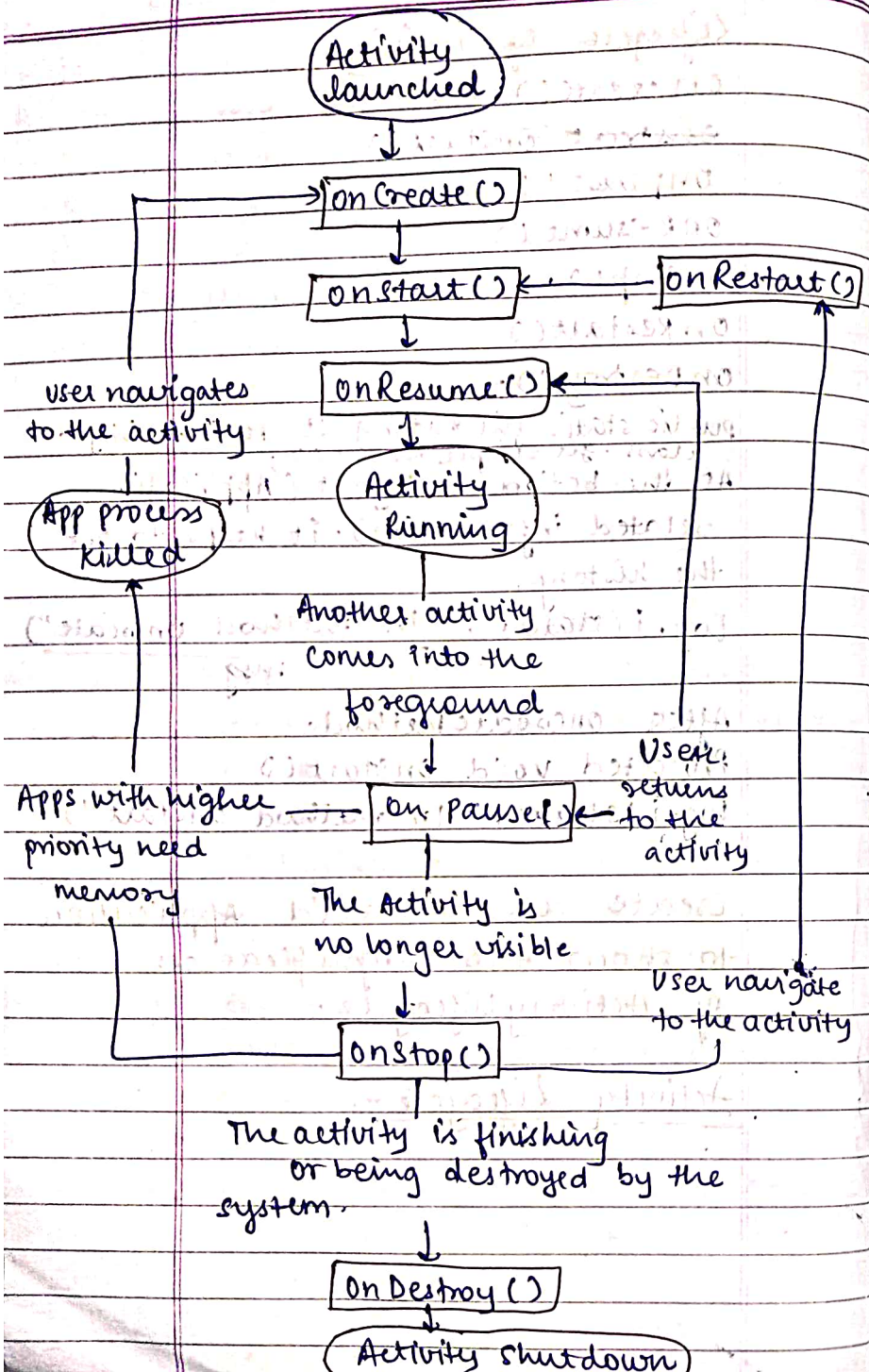
Date : / /
Page :

Lifecycle of Activity -
onCreate().
~~onStart~~ OnStart()
onPause()
onResume()
onStop()
onRestart()
onDestroy().
public static final String st = MainActivity1.class.getSimpleName();
At the bottom → logcat (Application related information). it helps to get the history.
Log.i (status ; "In method onCreate") msg;

After onCreate Method.
Protected void OnStart()
log.i (status, "In method OnStart");

Create an Android Application to show the significance of Activity Lifecycle.

Activity Lifecycle -



* android : gravity = "center".
put it in layout if whole page content needs to be there in center.

App Component : Activity -
Android OS is a linux based system where all the applications running on the android device are linux processes and each applications lifecycle is pre-determined from start to finish. Activity are the most fundamental building blocks of an Android Application.
The android OS manages the main Activity to start the application. Therefore, the main activity is the first activity user can view.
Android Activities have been defined under the lifecycle to manage applications runtime from launch to the end of the application's life.

Activity lifecycle includes the following stages.

1) onCreate()-
This method is autogenerated by android studio itself. It is used to call the first activity created.

2) OnStart()-
This method is called when activity is becoming visible to the user.

3) OnResume()-
This method is called when the activity will start interacting again with the end user.

4) OnPause()-
This method is called when activity is not visible to the user.

5) OnStop()-
This method is called when the activity gets stopped and is no longer in the interactive mode with the end user.

6) OnRestart()-
This method is called after the activity is stopped prior to start.

7) OnDestroy()-
This method is invoked before the activity is destroyed.

App component - Broadcast Receiver-
Broadcast receiver: simply responds to the broadcasted message from other applications or from the system itself. These msgs are also known as intends or events. For eg- an Application can also initiate broadcast msg to let other applications know that some data has been downloaded on to the device and is available for them to use.

There are two important steps while creating a broadcast class.

1) creating a broadcast receiver
public class abc extends BroadcastReceiver {

2) Registering broadcast receiver
<applications>
<Receiver android:name="abc">
<intent-filter>
<action android:name=
"android.intent.action.
BOOT_COMPLETED"> ~~bootcompleted~~"
</action>
</intent-filter>
</Receiver>
</applications>

manifest file.

There are two types of Broadcast Receiver -
1) Static Broadcast Receiver
2) ~~Receiver~~ Dynamic Broadcast Receiver

Static -
They are declared in the manifest file and works even if the application is closed.

Dynamic -
They worked only if the application is active or minimised.

App Component -

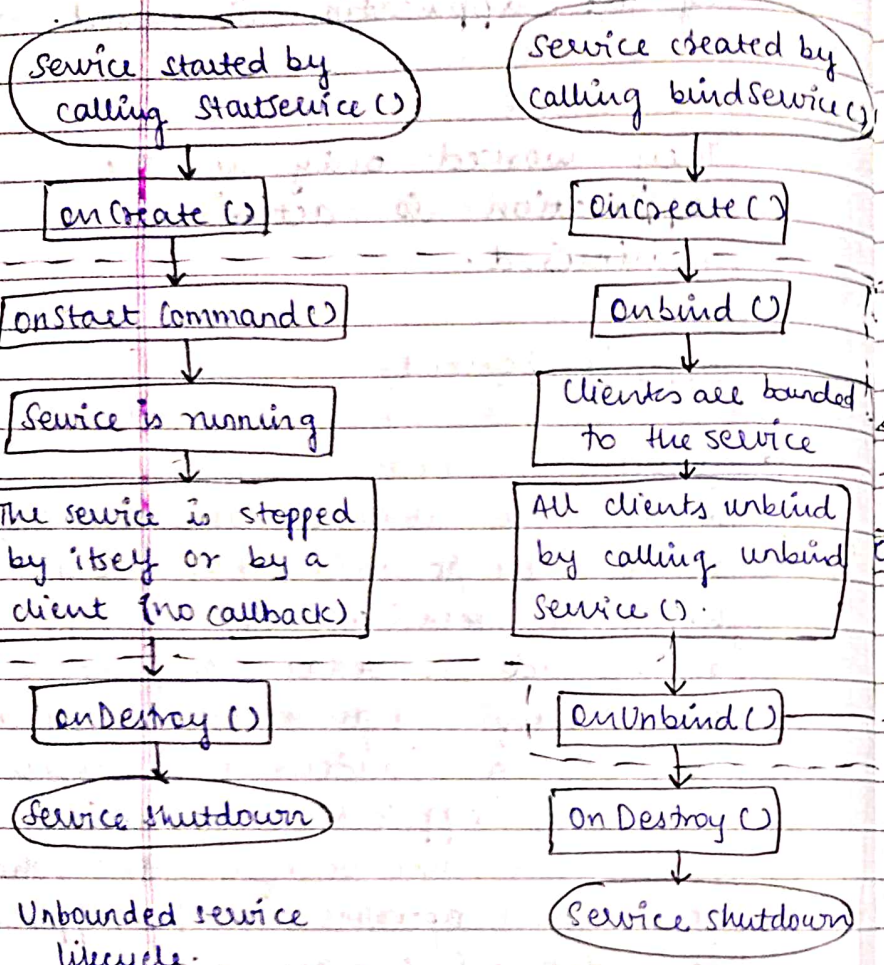
- Services

Services in android studio are special components that facilitate an application to run in the background in order to perform long running processes. The prime aim of a service is to ensure that the application remains active in the background. so, that user can operate multiple Applicat at a time. A service can run continously in the background ever if the application is closed or the user switches from one Application to another their is a major difference b/w service and a thread. Thread is a feature provided by operating system an

Local bind → IBinder()
Hybrid → Message API

Date: / /
Page:

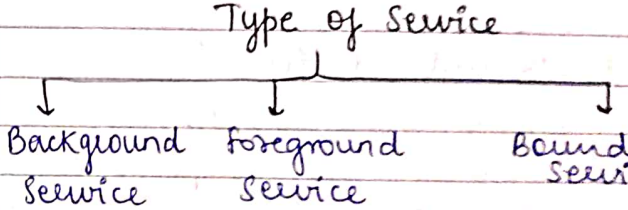
Diagram -



a service runs in the background to perform long running processes.

for eg- of android service :-
playing music in the background

Types of Services -



1) **Foreground Service** -
Services that notify the user about it ongoing operations are termed as foreground services user can interact with the services by the notification provides about the ongoing task such as downloading a file, playing music, etc.

2) **Background Service** -
Background Services do not require any user interaction.

These services do not notify the user about the ongoing background task and user cannot also access such task.

for ex:
Scheduling of syncing the data, storing of the data, etc.

3) Bound Service-
These services of android allows the component of the application to bound themselves with each other Bound services perform their task as long as any application component is bound to bind themselves.

~~Bind~~ bindService method is invoked to bound the components of the applications.

Two types of bound services
- Local
- Hybrid

Assignment-
History of Android:
The history and versions of andro are interesting to know. The code names of android ranges from A to J currently, such as Aestro, Blender Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice cream Sandwich, Jelly Bean, Kitkat and Lollipop. Let's understand the android history in a sequence.

1) Initially, Andy Rubin founded Android Incorporation in Palo Alto, californi United States in October, 2003.

2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.

3) The key employees of Android Incorporation are Andy Rubin, Rich Miner, Chris White and Nick Seaes.

4) Originally intended for camera but

shifted to smart phones later because of low market for camera only.

5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots.

6) In 2007, Google announces the development of android OS.

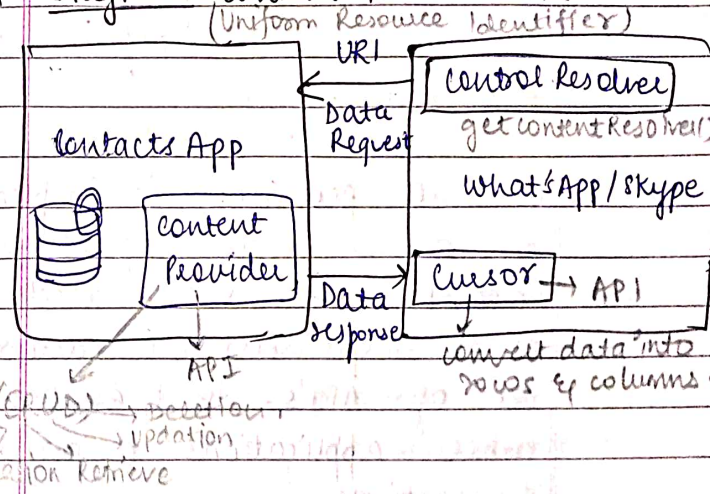
7) In 2008, HTC launched the first android mobile.

Android versions, codename and API

Version	Codename	API level
→ 1.5	Cupcake	3
→ 1.6	Donut	4
→ 2.1	Eclair	7
→ 2.2	Froyo	8
→ 2.3	Gingerbread	9 and 10
→ 3.1 and 3.3	Honeycomb	12 and 13
→ 4.0	Ice cream Sandwich	15
→ 4.1, 4.2 & 4.3	Jelly Bean	16, 17 & 18.
→ 4.4	Kitkat	19
5.0	Lollipop	21
6.0	Marshmallow	23
7.0	Nougat	24-25
8.0	Oreo	26-27

→ App component - Content Providers -

* Diagram - Content Provider - Android OS



- Content Provider are very imp. components that serves the purpose of a relational database to store the data of an application. The role of content provider in android is like a central depository in which data of the applications

are stored and it facilitates other applications to securly access and modifies the data based on the user requirement.

- In order to share the data, content provider have certain permissions that are used to grant or restrict the rights to other applications to interfere with the data.

- Content provider is implemented as a sub class of ~~Content~~ ContentProvider class and must implement a standard set of API's, that enable other applications to perform transactions.

- Four fundamental operations that are possible in Content Provider are

1) Create -
This operation is used to create the data in a contentProvider.

2) Read -
It is used to fetch the data from a content Provider.

3) Update -
This method is used to modify the existing data.

4) delete -
This is used to remove the existing data from the Content Provider.

Content URI -
Content URI or Content Uniform Resource Identifier is the key concept of content provider. To access the data from the Content Provider, URI used as a query ~~spe~~ string.

Structure of Content URI -
Content : // Authority / optional Path / optional ID.

1) Content : // - this represents that the given URI is a content URI.

2) Authority / -
It signifies the name of the content Provider like contacts, Browser, etc. This must be unique for every Content Provider.

3) Optional Path / -
It specifies that the data type provided by the content provider like audio files, or video files.

4) optionID -
It is a numeric value that is used when there is a need to access a particular record.

Create an Android Application to take input from the user and print the sum on the text box.

Create an Android Application to depict basic calculator.

→ Relative Layout
It is a view group that display child view in a relative position.
Preposition The position of each view can be specified as relative to sibling element.
For eg -
Component can be set left, below, right, above to the sibling element.

* coding - input -

<Relative layout > - - - - - >
Padding = 15 dp.
<EditText
height = match Parent
input_type = "number"
id = "@+id/ET1" >
layout_centerhorizontal /> .

<EditText
layout_width → height → match Parent wrap parent cont
layout_below = @+id/ET1
margin = "30dp" → space blw two cell.
android id = @+id/ET2.
input type = "number" />

<TextView
~~<appcompat~~
layout_width =
layout_height =
layout_below = @ id/ ET2
id =
text = "0"
textsize = "55dp"
centervertical = "true"
centerhorizontal = true />

<Button
width
height.
id =
text = "ADD"

Java -
Button b1
EditView Et1, Et2
TextView tV1;

findview by id - - -

b1.SetOnClickListener(new viewset
onClickListener - - - -
@ override
~~protected~~ ~~view~~ onClick
public void

if (Et1.getText().tostring().length()==0)
{Et1.setText("0")}
if (Et2.getText().tostring().length()==0)
{Et2.setText("0")}

int n1 = Integer.parseInt(Et1.getText
.tostring());

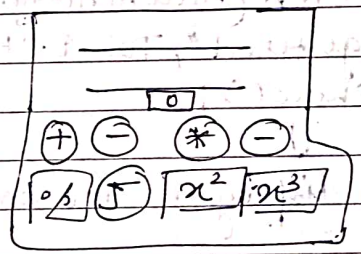
int n2 = Integer.parseInt(Et2.getText
.tostring());

int sum = n1+n2;
tV1.setText(String.valueOf(sum));

~~Image~~ ~~View~~

→ Toast - pop-up messages.

Create an android Application to create simple calculator.



XML
<Button
width
height
id = "b1"
/>

Date : / /
Page :

Java

```
Button btn;
btn.setOnClickListener(new
        View.set) - - ;
        - - -
        - - -
public void OnClick -
```

* Toast.makeText(Context; "msg";
  length long (5-10) — "Time").
  length_short. (1-2)
  this, getApplication Context.
  (Konsi current activity m karre ho) (puri Application ka context pass kardega)

```
Toast.makeText(MainActivity.this,
"User registered", Toast.LENGTH_
LONG).show();
```

↓

display on the Emulator.

OR

```
Toast.makeText(getApplication
Context(), "msg", Toast.LENGTH_
SHORT).show();
```

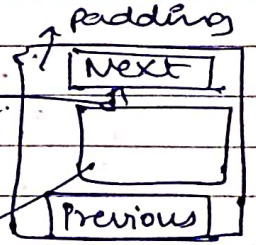
Date : / /
Page :

Create an Android Application to design the following XML Activity.

Name _____
Age _____
Add _____
Ph no. _____
[Submit] → Red.

Generate a toast msg when submit msg is pressed.

→ Image View = margin for space.



XML -

```
<Relative layout ... -
- - -
```

container to set img

```
<Image View
    height = . 190 dp
    width - . 280 dp.
    id = " "
/>
</Relative layout>.
```

Date: / /
Page:

Java
ImageView IV;
Button btn;
find View by id.

btn.setOnclick Listener
—{— — —
IV1.setImage Resource
(R.drawable.picture_name);
}
btn2.setOnClickListener:
{
IV1.setImageResource.
(R.drawable.picture_name2);
}

How to Insert in Drawable the picture.

App
Res
layout
Drawable → right click
file pasted.

Paste

Date: / /
Page:

Create an Android Application to change the image on an Activity on Button Click.

View class -
View is the base class for all the weight which appear in the view part of the emulator. All the class inlisted in the view class, must extends the view. The custom view can be directly calling sub class of view class like Button, TextView, EditText, listView, imageView, gradeView, etc.

Toast -
xml → <Button>
java → btn.setOnClickListener(new - View.set);
Toast.makeText - - -
— — — — -

Toast is a pop up message which display some information about an action peform in the activity area. The

pop up message is displayed for a short period of time.

Syntax-
Toast.makeText(context, "msg", time);

Eg- Toast.makeText(MainActivity.this, "Button Pressed", Toast.LENGTH_LONG).show();

Context -
It is about the surrounding information, it is very important to understand the environment.

Types of Context -

2 types of context.
① Application context
② Activity context.

Application Context -
This context is tied to the life cycle of an application. It is an instance i.e. single term and can be accessed using getApplicationContext();

It is necessary to create a singleton object.

getApplicationContext() is used to write the context of an application which holds all the activity running inside it.

When we call a method or a constructor we often pass a context using this keyword or getApplicationContext().

Activity -
In this each and every activity on screen get an activity. This method invoke the activity context of getContext().

getContext() method return the context which is link with activity from which is called. This is useful when we need to call the context from the current running activity.

Array Adapter-

* Used to creating linking between two files.
  XML - ListView & TextView
  Java.

```
Protected void onCreate(Bundle
  savedInstanceState){
    super.onCreate(SavedInstaneState);
SetContentView(R.layout.activity_
  main);
ll = findViewById(R.id.ll);
String list[] = {"C", "Python", "Java"};
ArrayAdapter<String> ab =
new ArrayAdapter<String>
(context-this, R.layout.list, R.id.tv,
list);
ll.setAdapter(ab);
}
```

Create an Android Application to design the following XML file and ~~bind~~ show the concept of ArrayAdapter.

Name	Rollno

→ Native and Hybrid Applications-

Native Applications are those applications that are designed for smart phones and are specially reliable on Mobile operating systems. For example swift for iOS., java/ kotlin for Android Mobile Applications.

Advantages of Native Application.

1) We can download and install these Applications on the device from google playStore or the Appstore. These Applications are accessed by the icons on the homescreen.

2) They are faster and more reliable to the end user.

3) They take advantage of Mobile features such as GPS, camera, compass, contactlist, etc.

4) They can work offline by using notification system of the device. It can be done by using push notification and by alerting the users everytime there is new piece of content published or if user's attention is required.

Disadvantages of Native Applications

1) They are developed on a single platform and will not run on any other platform.

2) A large amount of budget is required to develop a native Application that will be compatible with multiple platforms

3) Updation of the Application can be done only on google playstore or the Appstore.

4) Maintainance of the Native Applicat is higher.

Hybrid Application are the combinati of native Applications and web Applications. Hybrid App's runs offline on the device. They are developed using javascript, XML, HTML5, CSS3, etc. Hybrid App. are designed in a way to render existing websites content in an Application. Hybrid Apps can be accessed from a web browser or can be downloaded from an Appstore. The speed of Hybrid Applications is entirely dependent on the performance of the web Browser Used.

Date: / /
Page:

Advantages of Hybrid App.

1) Hybrid App. can work on multiple platforms.

2) The maintainance and development of Hybrid App. is slightly slower as compared to native App.

3) Hybrid App. are comparatively cheaper to develop.

Disadvantages of Hybrid App.

1) Hybrid App. require fast and uninterrupted internet speed.

2) They are slower than native App.

Difference b/w Native and Hybrid Application.

→ Booting Process
→ Lifecycle.

Date: / /
Page:

Native	Hybrid.
• are compatible with a single platform.	• can run on a multiple platform
• requires installation before use.	• do not require th installation for th use.
• maintainence is higher.	• Maintainance is lower.
• large amount of budget is required to develop. i.e. compatible with multiple platforms.	• low budget is required for the development.
• They have multiple code bases.	• They have a single code base.
• They provide good comfort & are reliable.	• They are less reliable and less user comfortable.

Date : / /
Page :

- They can be developed by using swift, java, kotlin.
- These App are developed by using HTML 5, CSS 3, javascript, JQuery, angular, XML, etc.

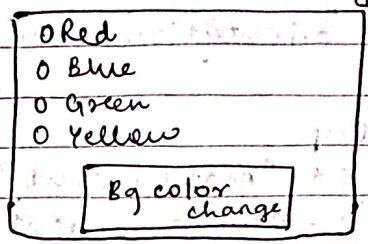
View : Radio Buttons -

```
<Linear Layout
android : orientation = horizontal / vertical />

<Radio Group
        height = " "
        width = " "
        id = " " />

<RadioButton
        height = " "
        width = " "
        text =
        textsize = " " />
</Radio Group>
```

Create an android application to take four radio Buttons of different colors. When user select any radio button

Date : / /
Page :

background color of text view should be changed.



Solution -

```
protected void OnCreate (Bundle saved
Instance state) {
super.OnCreate (SavedInstanceState);
b1 = findViewById (R.id.b1);
b2 = findViewById (R.id.b2);
b3 = findViewById (R.id.b3);
b4 = findViewById (R.id.b4);

RG1.setOnclickListener {

case @@.id /R.B1/ -
        "  Set Backgroundcolor - - .
        break;
```

XML-

```
<LinearLayout
android:orientation = horizontal />

<RadioGroup
    height="match parent"
    width="wrap content"
/> id = "RG1" >

<RadioButton
    height = "match parent"
    width = "wrap content"
    text = "Blue"
    textsize = "50dp"
    id = "RB1" />

<RadioButton
    height = "match parent"
    width = "wrap content"
    text = "Red"
    textsize = "50dp"
    id = "RB2" />

<TextView
    height = "20dp"
    width = "50dp"
    id = "TV1" />
```

Android SDK-

To download and install latest android API's and development tools from the internet android provide us with android SDK Manager.

Android SDK manager seperates the API's, tools and different platform into different packages which the developer can download.

Running Android SDK Manager-

1. Click on tools and select SDK Manager option.
2. You can select the package you want to download and
3. click on the install button.
4. By default SDK Manager keeps itself up to date with latest API's and other packages.

→ Some important packages under SDK package are-

1) SDK TOOLS-

Necessary package to run your SDK.

2) SDK Platform Tools -
This package will be installed once you run the SDK Manager.

3) SDK platform -
Atleast one platform should be installed in your android environment to run the application.

4) System Images -
To download system images for all the ~~environ~~ android versions, so that you can test your applications on them with android emulator.

5) android samples -
This will give you some sample codes to learn about android.

→ Spinner - XML File -

```
<Relative layout>
<textview
    width =
    height =
    text =
    id = "t1"
    padding = "40dp"
    background = "@color/teal_700"
/>

<spinner>
    width = " "
    height = " "
    Alignment = "center"
    id = " s1 "
    padding = " "
    background = "@color/teal_200"
/>

</Relative layout>
```

JAVA File -

```
Spinner sp1;
textview tv1;
String s[] = {"ind", "PAK", "AUS"}
@ override
```

```
protected void - - -
 setContentView(R.layout.
  activity_main);
 sp1 = findViewById(R.id.s1);
 tv1 = findViewById(R.id.t1);
 tv1.setOnClickListener(new View.
  OnClickListener(){

@Override
  public void onClick(View view){
ArrayAdapter a = new ArrayAdapter
(this, android.R.layout.simple_
    - - - .

a.setDropDownViewResource(
 android.R.layout.simple_spinner_
 drop - - -

}
});
}
```

Toggle Button -

```
<Toggle Button
  - - -
  - - -
```

~~Java file~~

```
On Click Text="" =>
Off Click text="" => />
```

XML -

Toggle Button

Java

```
toggle button toggleButton1;
Button buttonSubmit;
 @override
  protected - - - .
  super.onCreate(savedInstanceState);
  - - - -

 toggleButton1 = find View by id
      (R.id.togglebutton);
buttonSubmit = find view by id
      (R.id.button);
buttonSubmit.setOnClickListener(
      new.View.OnClickListener();
public void - -
 String res;
 res = toggleButton1.getText().toString
Toast.makeText(getApplication
      Context(), res, Toast.LENGTH.
      LONG).show();
 }
```

AutoComplete TextView - XML

```
<textView>
<AutoComplete textView>
```

Java

```
String [] lang = {"c", "c++",
"Java", ".NET", "iphone", "Android",
"ASP.NET", "PHP"};

@Override
protected - -
  - - - -
  - - -

AutoCompleteTextView act = findView
 ById (R.id.autocompletetextview);
ArrayAdapter <string> adapter =
 new ArrayAdapter<string>
(context: this, android.R.layout.
select_dialog_item, lang);
act.setThreshold(1);
act.setAdapter(adapter);
}
```

View Class ; Toggle Button -

Android Toggle Button can be used to display check or unchecked state of a button. The component is beneficial if the user have to change the settings blw the two states. i.e. "On" and "off". Such components can be used while developing wifi, bluetooth, hotspots, etc.

Since Android 4.0 there is i.e. another type of toggle Button known as switch that provides slider control. Android toggle Button and switch are both the sub classes of CompoundButton class.

Some methods of toggleButton class are :

1) Charsequence getTextOff() -
It returns the text when the button is not in the checked state.

2) Charsequence getTextOn() -
It returns the text when the button is checked.

View Class : AutoCompleteTextView -

AutoCompleteTextView is a view i.e similar to Edit Text

except it shows a list of complete suggestions automatically while the user is typing.
The list of suggestions is displayed by dropdown menu. The user can choose an item from the dropdown menu and replace the content of EditBox.

Edit Text -

android:Hint = "Enter your name"

except it shows a list of complete suggestions automatically while the user is typing.
The list of suggestions is displayed by dropdown menu. The user can choose an item from the dropdown menu and replace the content of EditBox.

Edit Text -
android:Hint = "Enter your nam

View : Grid Layout -

XML → Relative layout.
background = "#FFEOCB".

<GridLayout
android:layout width = "match parent"
" " height = "
rowcount = "5"
columncount = "3"
padding = "5dp" >
<Button
don't use them { width = "~~match~~ wrap"
height = "wrap"

Spinner-x
grid layout-x
Background- onClick - textView, b.g., Button.
Activity lifecycle.
Radio button

Date / / Day
Page Add.

text = "Button"
Margin = "5dp"
layout_gravity = "fill"
- row = "1"
- rowWeight = "1"
- columnWeight = "1".
/>
<Button>
— — —
— — .
<Button>
— — — .
</Grid layout>.

Ques) Create an Android Application to demonstrate grid layout by implementing multiple image view tags. / image gallery-

import os.Bundle.
// importing all essential bundles.
public class mainActivity extens
— — —
- Overrea — — —