Mobile Programming BCA 6th Semester

June
7/1/2022

Mobile Programming (Unit 1,2,3)

① Steps to create New project in Android studio.

Ans:
→ Install latest version of android studio
→ Goto file >> New Project
→ Select Empty Activity
→ Give name of project & save location
→ Click finish

② View

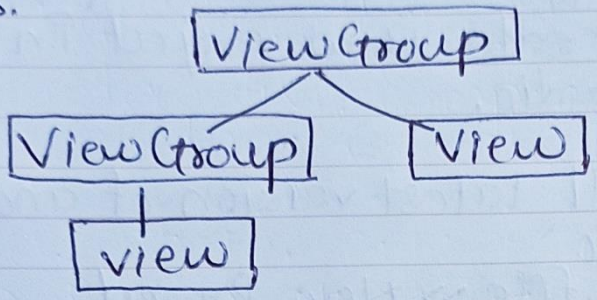
→ It is an user Interface seen on the screen.
eg. Button, checkbox, Edittext etc.
color, dimension & position are some attributes of view

③ Context

→ It is an Interface to global information about an application environment.
types:
Application → getApplicationContext()
Activity → this / getContext()

④ Hierarchy of viewgroups & views

Ans.



* View class are the basic building block for user interface components

* ScrollView: contains one element and enables scrolling

* RecyclerView: Contains a list of elements & enables scrolling by adding & removing elements dynamically.

⑤ Layouts

Ans: They are the special types of viewGroup which contains child views that can be a row, column, grid, table, absolute etc

⑥ Event Handeling

Ans. Events are useful way to collect user's data using interactive components of app. Eg. Btn, screen-touch etc.

ⓐ Event Listeners Registration

→ Process of registered an event handler to the event listener

ⓑ Event Listeners

→ It is an interface in the view class having single callback method it is fired when interaction is done to the view in which the listener was registered.

ⓒ Event Handler

→ they are the methods which is registered in event listener & triggered when an event occcer.

Event Handler	Event listener
onClick()	onClickListener()
onfocusChange()	OnfocusChang Listener

⑦ Android layout attributes

Ans: They are set on android layouts

eg. android:id
android:layout_height
android:layout_gravity

⑧ Android widgets

Ans: It is a small control of your app placed on homescreen.

eg. Textbox, Edittext, spinner etc)

⑨ Android Resources

→ It is an additional file and static content that your code uses, such as bitmaps, strings, layout definitions

Location: app >> res ---

- xml files are compiled resources
- Raw files (audio, video) are uncompiled resources

eg. <TextView android:text="@string/msg" />

Unit-4 June 9-2021

(10) Android Activity

Ans: Activity is a single screen with a user interface just like window or frame of Java. An application can contains more than one activity. but one file has one activity.

- It handles user interaction such as button click, text entry etc
- first activity that user sees is a main activity

layout of an activity is defined in XML file

Defining Activity java class

```
public class MainActivity extends
  AppCompactActivity {
@override
protected void onCreate(
  Bundle b)
{ super.onCreate(b);
}
}
```


connect activity with layout
→ setContentView(R.layout.activity_name)

Adding activity in manifest
→ <activity android:name="activity" />

* Use <intent-filter> to declare activity as main activity

(11) Activity Lifecycle

onCreate()
onStart()
onResume()
onPause()
onStop()
onDestroy()

(a) onCreate()
→ Called when activity is firstly created. We perform the actions that should happen only once for the entire life of activity.
eg. setLayout, initiate variables

(b) onStart()
→ called when activity is becoming visible and prepares activity to enter in foreground & become ~~interactive~~ interactive

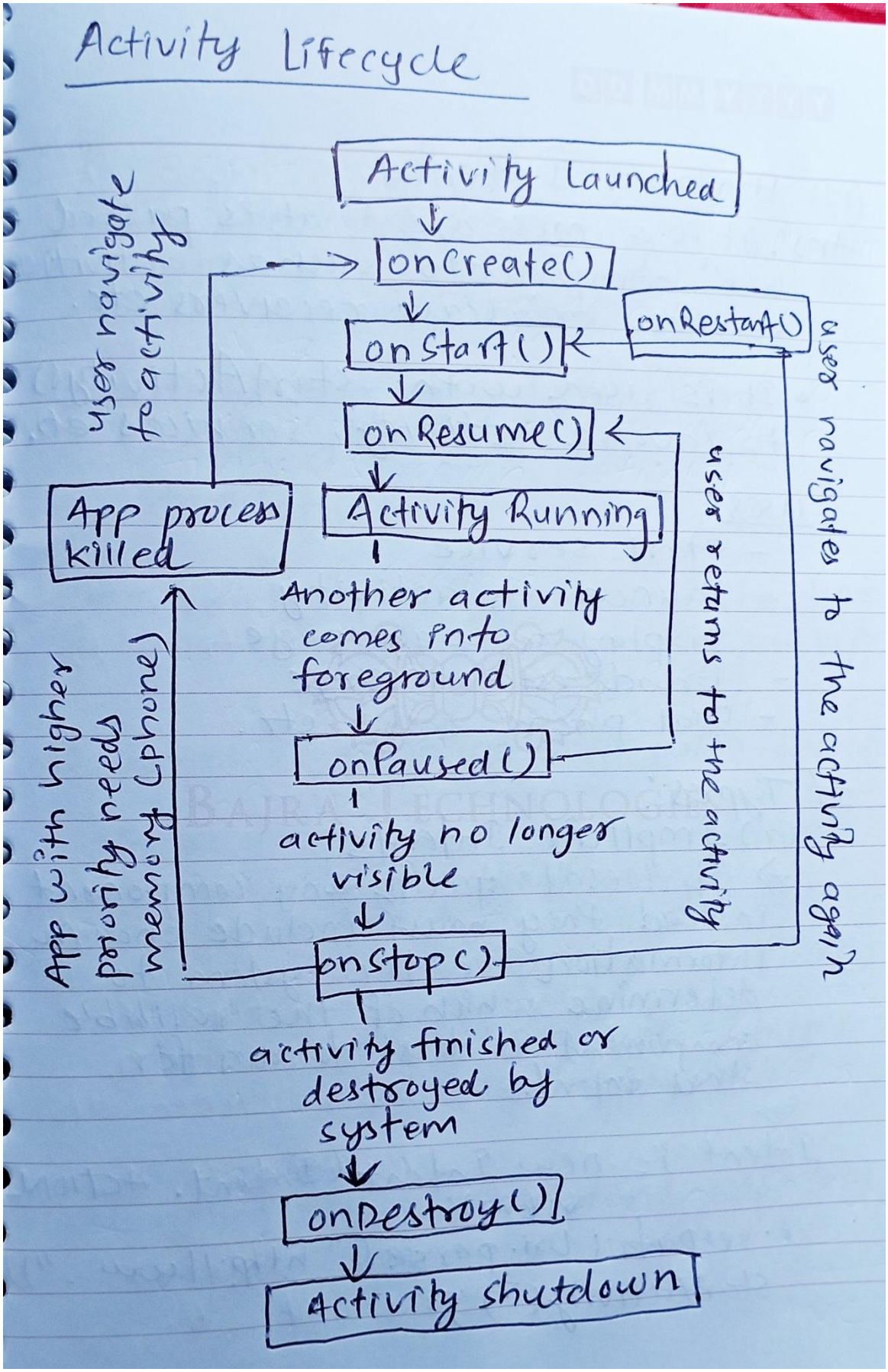
(c) onResume()
⇒ called when activity starts interacting with user. app stays in this state until something happens to take focus away from app.
eg. Incoming calls.

(d) onPause()
→ called when activity is not visible to user. Not necessary to destroy the app sometimes switch to another activity or app.

(e) onStop()
→ called when activity is no longer visible. Means the activity is covered by another activity on screen. Here all resources are released or adjusted if they are not needed.

(f) onRestart()
→ called if the activity comes back after calling onStop().

(g) onDestroy()
→ called if the app is completely closed or destroyed temporarily due to configuration change (screen rotate)



(12) Android Intents

Ans'. It is a message that is passed bet^{n} components such as activities, services, broadcast receivers etc.

- It is used with startActivity() to invoke activity, services etc.

Uses':

- start service
- launch an activity
- Display a web page
- Broadcast message
- Dial phone calls etc.

 # Types':

(a) Implicit Intents

→ It doesn't specify any component instead they must include enough information for the system to determine which of the available component is best to run for that intent.

Eg.

```
Intent i = new Intent(Intent.ACTION_
                VIEW);
i.setData(Uri.parse("http://ww_"));
startActivity(i);
```


(b) Explicit Intent

→ It specifies the component so it provides the external class to be involved.

eg:

```
Intent i = new Intent( this,
    ActivityB.class);
startActivity();
```

- Intent help to communicate bet^n android system and our activity.

Types of sending data with Intent

- Data - one piece of information whose data location can be represented by an URI

```
//sending
intent.setData(Uri.parse("http://
    www.notebahadur.com"));

//Receiving
Uri locationData = intent.getData();
```


- Extras - one or more pieces of information as a collection of key-value pairs in a Bundle.

```
// sending
putExtras (string name, in val)
intent.putExtras ("NUM", 2);

// Receiving
int i=intent.getExtra ("NUM", 0);
Bundle b = intent.getExtras ();
```

- use extras for bundle data extra for other while sending & receiving data.

(13) Android Manifist file

Ans It is an important file of our app which defines the structure & meta data of our application, its components & its requirements.

- summary of an application
- components includes
  - user permissions (camera)
  - Declaring API level etc.

Unit-5

fragments

July -4-2022

(14) About fragment

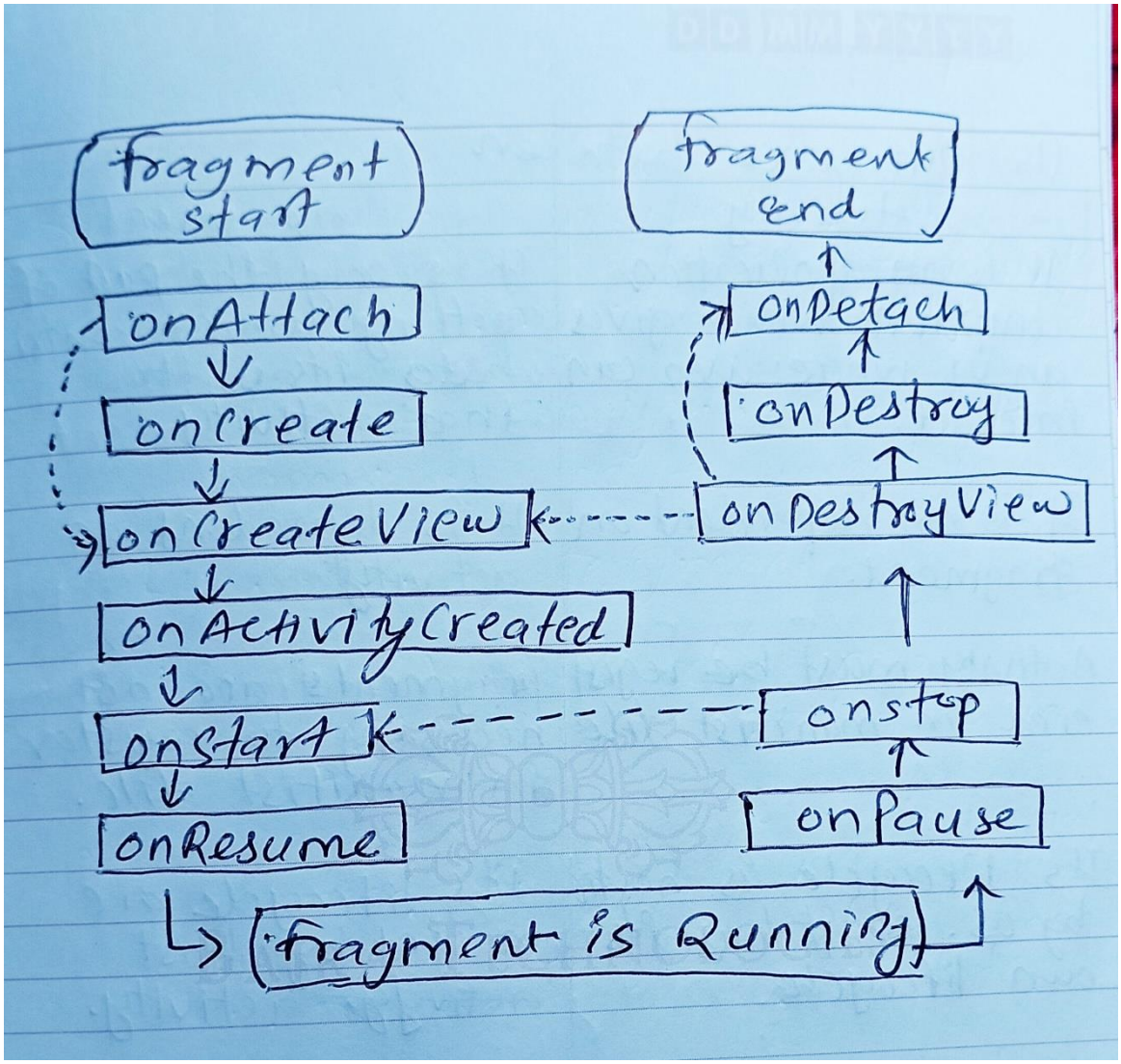
→ fragment is a reusable class which is dependent on ~~any~~ activity It is a portion of an activity.

fragment manager is used to dynamically add fragment in java.

Lifecycle of fragment

- onAttach() is called when a fragment is connected to an activity
- onCreate() is called to do initial creation of fragment
- onCreateView() is called when the fragment should inflate a view.
- onViewCreated() is called after onCreateView() which ensures that the fragment's root view is not-null.
- onActivityCreated() is called when host activity has completed its onCreate() method.
- onStart() is called when a fragment is ready to display on screen.

- onResume() is called when fragment become active & allocate resources (location, sensor)
- onPause() is called when user is leaving the fragment and it also releases expensive resources (registering for location, sensor update etc).
- onDestroyView() is called when fragment's view is being destroyed, but the fragment is still kept around.
- onDestroy() is called when fragment is no longer in use
- onDetach() is called when fragment is no longer connected to the activity.



(15) Difference between

Activity	Fragment
It is an application component which gives an UI where user can interact.	It is only the part of activity which contributes its UI to that activity.
It is not dependent on fragment.	It is dependent on activity
Activity must be registered in manifest file	Fragments are not necessary to register in manifest file.
Its lifecycle is hosted by OS. ie. It has its own lifecycle	Its lifecycle are hosted by host ~~activity.~~ activity.
they are not light weight.	they are light weight.

(16) Menu in android

→ Menu is a part of the UI component which is used to handle some common functionality around the application.

Types

(a) Option Menu
(b) Context Menu
(c) Popup Menu

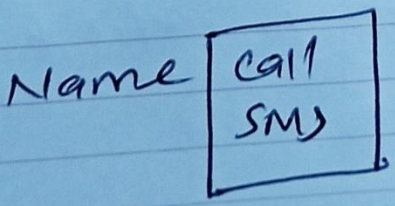
(a) Option Menu

→ It is a primary collection of menu items in an android application and is useful for the action that have global impact on the searching application.

(b) context Menu

→ It is a floating menu that only appears when user clicks for a long time on one element & it only affects the selected content or context frame.

Eg.



(c) Popup Menu

→ It displays a list of items in a vertical list which presents the view that invoked the menu and is useful to provide an overflow of actions related to specific content.

(17) Dialog

→ It is a small window that prompts the user to make a decision or enter additional information.

Types:

Alert Dialog
DatePicker Dialog
TimePicker Dialog
Custom Dialog

Chapter-6 (List view & Recycler view)

(18) ListView

→ ListView is a view which groups several items and display them in vertical scrollable list. Here, the list items are automatically inserted to the list using an Adapter that pulls content from source such as an array or database.

steps to use ListView
- Add listview to your Layout
- create array & array Adapter
- Attach adapter to listView

(19) GridView

Ans: A gridview is a type of Adapter view that displays items in a two-dimensional scrolling grid.

(20) RecyclerView

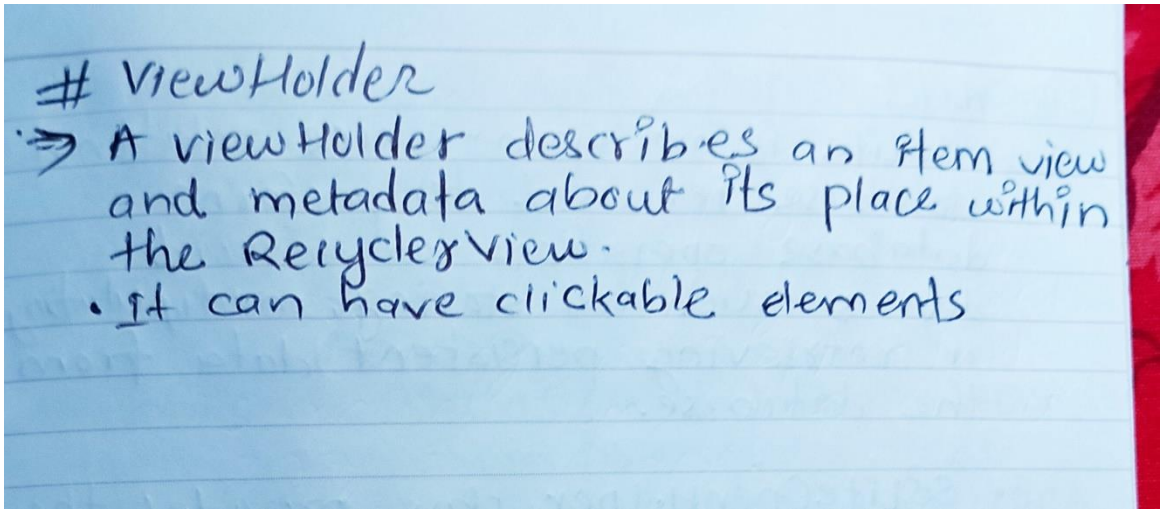
Ans: RecyclerView is an advance version of listView with improved performance. It extends ViewGroup class and scrollingView interface.

features of RecyclerView
- Advance & flexible version of listview which is used to display large dataset.
- contains integrated animations
- enforces recycling of views.
- supports both grid and lists.
- supports, it supports both vertical & horizontal scrolling.

Layout Manager
⇒ It is used to position view items inside a RecyclerView, also it reuses view items that are no longer visible to the user.

Buil-in layout managers:
- LinearLayoutManager
- GridlayoutManager
- StaggeredGridLayoutManager

Adapter
⇒ It is an intermediary between data and view which manage creating, updating, adding, deleting view items.



Unit-7 Advance android (DB)

(21) SqLite

Ans: SQLite is an open-source relational database ie. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database.

- SQLiteOpenHelper class provides the functionality to use the SQLite database.

#Advantages of SQLite DB

→ Light weight

→ Better performance : it only loads required data rather reading all.

→ Reliable: content can be updated continuously and automatically, less bugs as SQL queries are smaller than procedural code.

→ Portable : The application file is portable across all OS (32/64 bits)

→ Accessibility : contents can be viewed using a wide variety of 3rd-party tools and its data is recoverable in future.

→ Reduced cost and complexity

Difference Betn

SQLite	SQL
Written in ANSI C	Written in C
Embeddable RDBMS	used to query RDS
It is file based. Doesn't have a seperate server process embedded in client.	standard. specifies how a relational schema is created and many more.
supports many features of SQL. High performance.	It is not a database itself.
Portable database resource.	Structured Query Language
Uses SQL.	Used by different SQL databases like oracleDB, MySQL

Disadvantages of sqlite.
- Lacks of multi-user capabilities
- Limited storage.
- Bottleneck when storing large dataset, as it is file based. (slow data transfer)

SQLiteOpenHelper class

```
public class DatabaseHelper extends
  SQLiteOpenHelper {
private static String DB_NAME = "Test";
private static int VERSION = 1;

// constructor
public DatabaseHelper ( Context context)
  {
   super(context, DB_NAME, null,
       VERSION);
  }

@override
  public void onCreate ( SQLiteDatabase
    sqliteDatabase) {
String query = " CREATE TABLE Test" +
" (id INTEGER PRIMARY KEY," +
" name TEXT, age INTEGER)";
SqliteDatabase. ~~execute~~ execSQL(query);
 }
```



```
@Override
public void onUpgrade(SQLiteDatabase,
 int oldVersion, int NewVersion)
{
 sqliteDatabase.execSQL("DROP TABLE"+
"IF EXISTS test_table");
 onCreate(SqliteDatabase);
}
}
```

Cursors

→ Cursor is an object interface that provides random read-write access to the result set returned by a database query.

content values

⇒ It is a maplike class that matches a value to a string key.

eg.

```
contentValues cv = new contentValues();
cv.put("COL-FIRST", customer.getName());
```


insert(String table, String nullColumnHack, ContentValues values)

rawQuery(String sql, String[] selectionArgs)

(22) API (Application Programming Interface)

→ An API is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software.

OR

It is a mechanism that allows the interaction bet^n 2 applications using a set of rules.

Types:

ⓐ Public API
ⓑ Partner API
ⓒ Internal API
ⓓ Composite API

API protocols & Architecture

(a) SOAP (Simple object Access Protocol)

→ It defines messages in XML format used by web application to communicate eachother.

→ It is highly structured, tightly controlled & clearly defined standard.

eg:

```
<soap:envelope>
  <soap:body>
  </soap:body>
</soap-envelope>
```

(b) REST (Representational State Transfer)

→ It is a client-server architectural style that uses the http protocol in a simple and effective way.

(c) RPC (Remote protocol Call)

→ It is a simple means to send multiple parameters and receive results.

It supports 2 languages

JSON-RPC

XML-RPC

JSON
→ Javascript Object Notation is a format for ~~storing~~ structuring data.

advantages of JSON
- It is Human-readable and writable
- It is light weight text based data interchange format simpler to read and write.
- Language Independent

Volley
→ It is a library that makes networking for Android apps easy & faster.

It has 2 classes
RequestQue & Request that we have to deal with.

(23) Implementation of Google map

steps:

(a) Register your application in Google Developer console and enable API

(a) Open Google Developer console & sign in with gmail account

(b) Create New project

(c) Click APIs & services and open Dashboard

(d) Click Enable APIs and services

(e) search 'Google Map Android API' and enable it

(f) Refresh page and goto credentials

(g) Click on create credentials and choose API key. It will create API key to integrate maps in your application

(h) Save API key somewhere and use when implementing in your project.

Implementation in Application.

(a) Add <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/> in your manifest file.

(b) Add your generated key in manifest file inside application tag.

<meta-data android:name="key" />

(c) Add following dependency in build.gradle file.

```
dependencies {
    implementation 'com.google.android.gms:play-services-maps:15.0.1'
}
```

(d) Finally create an Activity by implementing OnMapReadyCallback interface.