Activity Life Cycle

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getMessage("onCreate");
    }

    @Override
    protected void onStart() {
        super.onStart();
        getMessage("onStart");
    }

    @Override
    protected void onResume() {
        super.onResume();
        getMessage("onResume");
    }

    @Override
    protected void onPause() {
        super.onPause();
        getMessage("onPause");
    }
```

```
    @Override
    protected void onStop() {
        super.onStop();
        getMessage("onStop");
    }

    @Override
    protected void onRestart() {
        super.onRestart();
        getMessage("onRestart");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        getMessage("onDestroy");
    }

    private void getMessage(String msg) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
    }
}
```

Telephony Api Demo

- **For accessing Telephone related data :**

```
ActivityCompat.requestPermissions(this, new String[]{READ_PHONE_STATE,
CALL_PHONE}, 100);
TelephonyManager tm = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
int netType = tm.getNetworkType();
switch (netType) {
   case NETWORK_TYPE_GSM:
      data += “Network = 2g or lesser\n”;
      break;
   case NETWORK_TYPE_HSDPA:
   case NETWORK_TYPE_HSPA:
   case NETWORK_TYPE_HSUPA:
   case NETWORK_TYPE_HSPAP:
      data += “Network = 3g\n”;
      break;
   case NETWORK_TYPE_LTE:
      data += “Network = 4g\n”;
      break;
   default:
      data = “Network = 5g\n”;
      break;
}
data += “Network Operator : “+tm.getNetworkOperator()+
      “\nNetwork Operator Name : “+tm.getNetworkOperatorName()+
      “\nDevice Software Number : “+tm.getDeviceSoftwareVersion()+
      “\nNetwork Country Iso : “+tm.getNetworkCountryIso();
t_data.setText(data);
```

- **For Calling through the application :**

```
call.setOnClickListener(v -> {
   Intent i = new Intent(Intent.ACTION_CALL);
   i.setData(Uri.parse("tel:+91"+c_no.getText().toString()));
   startActivity(i);
});
```

Notification Services Demo

- **Java Code**

```
NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

NotificationChannel channel = null;

Notification.Builder builder;

if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

    channel = new NotificationChannel("1", "testApp",
    NotificationManager.IMPORTANCE_HIGH);

    builder = new Notification.Builder(this, "testApp");

    manager.createNotificationChannel(channel);

    builder.setContentText("This is notification body..")

            .setContentTitle("This is Title")

            .setSmallIcon(R.drawable.ic_launcher_background)

            .setChannelId("1")

            .setAutoCancel(true);

} else {

    builder = new Notification.Builder(this);

    builder.setContentText("This is notification body..")

            .setContentTitle("This is Title")

            .setSmallIcon(R.drawable.ic_launcher_background)

            .setAutoCancel(true);

}

manager.notify(1, builder.build());
```

- **Kotlin Code**

```
val nm = getSystemService(NOTIFICATION_SERVICE) as NotificationManager
if(Build.VERSION.SDK_INT>=Build.VERSION_CODES.O){
val ch = NotificationChannel("CH1" ,"CUSTNOTI",NotificationManager.IMPORTANCE_HIGH)
nm.createNotificationChannel(ch)
val builder:Notification = NotificationCompat.Builder(this)
        .setContentTitle("Title")
        .setContentText("Content Text")
        .setSmallIcon(R.drawable.ic_launcher_background)
        .build()
        nm.notify(1,builder)
} else {
val builder = NotificationCompat.Builder(this)
        .setContentTitle("Titel")
        .setContentText("Content Text")
        .setSmallIcon(R.drawable.ic_launcher_background)
        nm.notify(1,builder.build())
}
```

Shared Preferences Demo

Following code stores data available in EditText and retrieves data from previously stored preferences

- **Saving Data**

```
SharedPreferences sp = getSharedPreferences("SPDATA",MODE_PRIVATE);
SharedPreferences.Editor ed = sp.edit();
ed.putString("Data",editText.getText().toString());
ed.apply();
```

- **Fetching Data from previously stored shared preferences**

```
SharedPreferences sp = getSharedPreferences("SPDATA" ,MODE_PRIVATE);
editText.setText(sp.getString("Data",""));
```

Tween Animation Demo

- **Alpha Animation XML:**

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="2000"
   android:fillBefore="true">

   <alpha
      android:fromAlpha="0"
      android:toAlpha="1"/>

</set>
```

- **Scale Animation XML:**

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="2000"
   android:fillAfter="true"
   android:repeatMode="reverse">

   <scale
      android:fromXScale="0dp"
      android:fromYScale="0dp"
      android:pivotX="50%"
      android:pivotY="50%"
      android:toXScale="100dp"
      android:toYScale="100dp" />

</set>
```

- **Rotation Animation XML:**

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="2000"
   android:fillBefore="true"
   android:repeatMode="reverse">

   <rotate
      android:pivotY="50%"
      android:pivotX="50%"
      android:fromDegrees="0"
      android:toDegrees="360"
      />
</set>
```

- **Translate Animation XML:**

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:fillAfter="true"
   android:repeatMode="reverse"
   android:duration="2000">

   <translate
      android:fromXDelta="0dp"
      android:fromYDelta="0dp"
      android:toXDelta="150dp"
      android:toYDelta="200dp"
      />
</set>
```

- **Call Animation From Java Code File:**

  Animation anim = AnimationUtils.loadAnimation(this,R.anim.animFileName);
  widget.startAnimation(anim);

Internal Storage Demo

- **Store in Internal Files Directory**
    - **Writing Data To File:**

```
try{
    FileOutputStream fos = openFileOutput("Data",MODE_PRIVATE);
    fos.write(editText.getText().toString().getBytes());
    fos.close();
}catch (Exception e){
    Log.e("MYError",e.getMessage());
}
```

    - **Reading Data From File:**

```
try{
        StringBuilder data = new StringBuilder();
        int d ;
        FileInputStream fis = openFileInput("Data");
        while((d=fis.read())!=-1){
           data.append(String.valueOf((char) d));
        }
        tv.setText(data.toString());
        fis.close();
      }catch (Exception e){
        Log.e("MYError",e.getMessage());
      }
```

- **Store in Internal Cache Directory**
    - **Writing Data To File:**

```
try{
        File fl = getCacheDir();
        File f1 = new File(fl,"MyFile");
        FileOutputStream fos = new FileOutputStream(f1);
        fos.write(editText.getText().toString().getBytes());
        fos.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

    - **Reading Data From File:**

```
try{
        File fl = getCacheDir();
        File f1 = new File(fl,"MyFile");
        data.append("\nPublic Cache Storage : ");
        FileInputStream fis = new FileInputStream(f1);
        while((d=fis.read())!=-1){
                data.append(String.valueOf((char) d));
        }
        fis.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

External Storage Demo

- **Store In External Public Directory:**
    - **Save Data:**

```
try{
    File fl = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
    File f1 = new File(fl,"MyFile");
    FileOutputStream fos = new FileOutputStream(f1);
    fos.write(editText.getText().toString().getBytes());
    fos.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

    - **Retrive Data:**

```
try{
    File fl = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS);
    File f1 = new File(fl,"MyFile");
    data.append("\nPublic External Storage : ");
    FileInputStream fis = new FileInputStream(f1);
    while((d=fis.read())!=-1){
        data.append(String.valueOf((char) d));
    }
    fis.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

- **Store In External Private Directory:**

    - **Save Data:**

```
try{
        File fl = getExternalFilesDir("MyDirectory");
        File f1 = new File(fl,"MyFile");
        FileOutputStream fos = new FileOutputStream(f1);
        fos.write(editText.getText().toString().getBytes());
        fos.close();
}catch(Exception e){
    Log.e("MYError",e.getMessage());
}
```

    - **Retrive Data:**

```
try{
        File fl = getExternalFilesDir("MyDirectory");
        File f1 = new File(fl,"MyFile");
        data.append("\nPublic External Storage : ");
        FileInputStream fis = new FileInputStream(f1);
        while((d=fis.read())!=-1){
                data.append(String.valueOf((char) d));
        }
        fis.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

- **Store In External Cache Directory:**

    - **Save Data:**

```
try{
        File fl = getExternalCacheDir ();
        File f1 = new File(fl,"MyFile");
        FileOutputStream fos = new FileOutputStream(f1);
        fos.write(editText.getText().toString().getBytes());
        fos.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

    - **Retrive Data:**

```
try{
        File fl = getExternalCacheDir ();
        File f1 = new File(fl,"MyFile");
        data.append("\nPublic Cache Storage : ");
        FileInputStream fis = new FileInputStream(f1);
        while((d=fis.read())!=-1){
                data.append(String.valueOf((char) d));
        }
        fis.close();
}catch(Exception e){
        Log.e("MYError",e.getMessage());
}
```

SQLite Demo (JAVA)

- **Sqlite.java:**

```
public class sqlite extends SQLiteOpenHelper {

    Context context;

    public sqlite(Context context) {
        super(context, "AK", null, 1);
        this.context = context;
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE users(id INTEGER PRIMARY KEY
    AUTOINCREMENT , nm Text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS users");
    }

    public void insert(String nm) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("nm", nm);
        db.insert("users", null, cv);
    }

    public void update(String id, String nm) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("nm", nm);
        db.update("users", cv, "id=?", new String[]{id});
    }

    public void delete(String id) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete("users", "id=?", new String[]{id});
    }
```

```
    public Cursor show() {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.rawQuery("SELECT * FROM users", null);
    }
}
```

- **MainActivity.java:**

```
public class MainActivity extends AppCompatActivity {

    Button ins,up,del;

    TableLayout tbl;
    EditText id,nm;
    sqlite sql;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        id = findViewById(R.id.id);
        nm = findViewById(R.id.nm);
        tbl = findViewById(R.id.tbl);
        ins = findViewById(R.id.add);
        up = findViewById(R.id.up);
        del = findViewById(R.id.del);
        sql = new sqlite(this);
        show();
        ins.setOnClickListener(v -> {
            sql.insert(nm.getText().toString());
            show();
            clear();
        });
        up.setOnClickListener(v -> {
            sql.update(id.getText().toString(),nm.getText().toString());
            show();
            clear();

        });

        del.setOnClickListener(v -> {
            sql.delete(id.getText().toString());
            show();
            clear();
```

```
            });
        }
        public void show(){
            Cursor c = sql.show();
            tbl.removeAllViews();
            TableRow thr = new TableRow(getApplicationContext());
            TextView th1 = new TextView(getApplicationContext());
            TextView th2 = new TextView(getApplicationContext());
            th1.setText("Id");
            th2.setText("Name");
            th1.setPadding(20,20,20,20);
            th2.setPadding(20,20,20,20);
            thr.addView(th1);
            thr.addView(th2);
            tbl.addView(thr);
            while (c.moveToNext()){
                TableRow tr = new TableRow(getApplicationContext());
                TextView t1 = new TextView(getApplicationContext());
                TextView t2 = new TextView(getApplicationContext());
                t1.setText(c.getString(0));
                t2.setText(c.getString(1));
                t1.setPadding(20,20,20,20);
                t2.setPadding(20,20,20,20);
                tr.addView(t1);
                tr.addView(t2);
                tbl.addView(tr);
                tr.setOnClickListener(v1 -> {
                    id.setText(t1.getText());
                    nm.setText(t2.getText());
                });
            }
        }
        public void clear(){
            id.setText("");
            nm.setText("");
        }
}
```

SQLite Demo (KOTLIN)

- **Sqlite.kt:**

```
class Sqlite(context: Context) : SQLiteOpenHelper(context,"ATOMIK",null,1) {
    private val context = context
    override fun onCreate(db: SQLiteDatabase?) {
        db.execSQL("CREATE TABLE AK (id INTEGER PRIMARY KEY AUTOINCREMENT,nm
        text)")
    }
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db!!.execSQL("DROP TABLE IF EXISTS AK")
    }
    fun insert(nm:String?=null){
        val db = writableDatabase
        val cv = ContentValues()
        cv.put("nm",nm)
        db.insert("AK",null,cv)
    }

    fun update(id:Int?=null,nm:String?=null){
        val db = writableDatabase
        val cv = ContentValues()
        cv.put("nm",nm)
        db.update("AK",cv,"id=?", arrayOf(id.toString()))
    }

    fun delete(id:String?=null){
        val db = writableDatabase
        db.delete("AK","id=?", arrayOf(id))
    }
```

```
    fun show(): Cursor {
        val db = readableDatabase
        return db.rawQuery("SELECT * FROM AK", null)
    }
}
```

- **MainActivity.kt:**

```
class MainActivity : AppCompatActivity() {
        lateinit var ins:Button
        lateinit var up:Button
        lateinit var del:Button
        lateinit var show:Button
        lateinit var nm:EditText
        lateinit var id:EditText
        lateinit var tbl:TableLayout
        var sql = Sqlite(this)
        override fun onCreate(savedInstanceState: Bundle?) {
                super.onCreate(savedInstanceState)
                setContentView(R.layout.activity_main)
                ins = findViewById(R.id.ins)
                up = findViewById(R.id.up)
                del = findViewById(R.id.del)
                show = findViewById(R.id.show)
                nm = findViewById(R.id.nm)
                id = findViewById(R.id.id)
                tbl = findViewById(R.id.tbl)
                ins.setOnClickListener {
                        sql.insert(nm.text.toString())
                        show.callOnClick()
```

```
}
up.setOnClickListener {
        sql.update(id.text.toString().toInt(),nm.text.toString())
        show.callOnClick()
}
del.setOnClickListener {
        sql.delete(id.text.toString())
        show.callOnClick()
}
show.setOnClickListener {
        val cur = sql.show()
        tbl.removeAllViews()
        tbl = findViewById(R.id.tbl)
        val thr = TableRow(this)
        val th1 = TextView(this)
        val th2 = TextView(this)
        th1.text = "ID"
        th2.text = "Name"
        th1.setPadding(5,5,5,5)
       th2.setPadding(5,5,5,5)
        thr.addView(th1)
        thr.addView(th2)
        tbl.addView(thr)
        while(cur.moveToNext()){
                val tr = TableRow(this)
                val t1 = TextView(this)
                val t2 = TextView(this)
                t1.text = cur.getString(0)
                t2.text = cur.getString(1)
```

```
                    t1.setPadding(5,5,5,5)
                    t2.setPadding(5,5,5,5)
                    tr.addView(t1)
                    tr.addView(t2)
                    tr.setOnClickListener{
                id.setText(t1.text)
                nm.setText(t2.text)
            }
            tbl.addView(tr)
        }
    }
}
```

GeoCoder Class Demo

- **Following is an example for fetching current location and print it in textview using GeoCoderClass.**

- **Permissions:**

    - <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    - <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

- **Gradle Implementation:**

    - implementation 'com.google.android.gms:play-services-location:21.0.1'

- **MainActivity.java:**

```
public class MainActivity extends AppCompatActivity {
    TextView location;
    Button get_Location;
    List<Address> list;
    FusedLocationProviderClient fusedLocationProviderClient;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        get_Location = findViewById(R.id.btn_getLoc);
        location = findViewById(R.id.loc);
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
```

```
        requestPermissions(new
String[]{Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION}, 100);

}

get_Location.setOnClickListener(new View.OnClickListener() {

        @Override

        public void onClick(View v) {

        fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(getApplicationC
ontext());

        if
(ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(getApplicationContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

        return;

        }


fusedLocationProviderClient.getLastLocation().addOnSuccessListe
ner(location1 -> {

        Geocoder gc = new Geocoder(getApplicationContext(),
Locale.getDefault());

                try {

                        list =
                gc.getFromLocation(location1.getLatitude(),location1.
                getLongitude(),1);

                        location.setText(list.get(0).getAddressLine(0));

                } catch (IOException e) {

                        e.printStackTrace();
```

```
                    }
                });
            }
            });
        }
}
```

Fragment Demo

- **Layout_main.xml**


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_margin="10dp"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/fg1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:text="Show Frag 1"
        style="@style/ThemeOverlay.Material3.Button.TextButton.Snackbar"
/>
    <Button
        android:id="@+id/fg2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:text="Show Frag 2"

style="@style/ThemeOverlay.Material3.Button.TextButton.Snackbar"/>
</LinearLayout>

    <androidx.fragment.app.FragmentContainerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
        android:id="@+id/fg"
        tools:layout="@layout/fragment_1"/>


    </LinearLayout>
```

- **MainAvtivity.java**

```
public class MainActivity extends AppCompatActivity {d
    Button fg1,fg2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fg1 = findViewById(R.id.fg1);
        fg2 = findViewById(R.id.fg2);
        fg1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentManager fm = getSupportFragmentManager();
                FragmentTransaction ft = fm.beginTransaction();
                ft.add(R.id.fg,Fragment1.class,null);
                ft.commit();
            }
        });
```

```
        fg2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentManager fm = getSupportFragmentManager();
                FragmentTransaction ft = fm.beginTransaction();
                ft.add(R.id.fg,Fragment2.class,null);
                ft.commit();
            }
        });
    }
}
```