L. J Institutes of Engineering and Technology

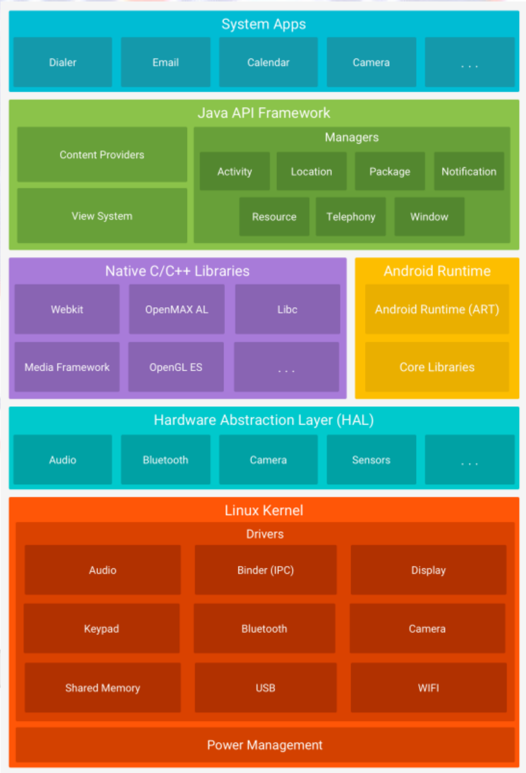
Remedial MSE List of Questions

SEM: 7
Subject Name: Mobile Application Development
Subject Code: 3170726

1 What is Android? Explain android architecture with diagram.

Android
Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers.

Android was developed by the Open Handset Alliance, led by Google, and other companies.



1) Linux kernel
It is the heart of android architecture that exists at the root of android architecture. Linux kernel is

	responsible for device drivers, power management, memory management, device management and resource access. 2) Native Libraries On the top of linux kernel, their are Native libraries such as WebKit, OpenGL, FreeType, SQLite, Media, C runtime library (libc) etc. The WebKit library is responsible for browser support, SQLite is for database, FreeType for font support, Media for playing and recording audio and video formats. 3) Android Runtime In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance. 4) Android Framework On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development. 5) Applications On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using linux kernal.
2	What are the core components of android explain in brief. Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact. 1.Activities They dictate the UI and handle the user interaction to the smart phone screen. 2.Services They handle background processing associated with an application. 3.Broadcast Receivers They handle communication between Android OS and applications. 4.Content Providers They handle data and database management issues. **Activities** An activity represents a single screen with a user interface,in-short Activity performs actions on the screen. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched. An activity is implemented as a subclass of Activity class as follows −

```
public class MainActivity extends Activity {
}
```

Services

A service is a component that runs in the background to perform long-running operations. For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity.

A service is implemented as a subclass of Service class as follows −

```
public class MyService extends Service {
}
```

Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

A broadcast receiver is implemented as a subclass of BroadcastReceiver class and each message is broadcaster as an Intent object.

```
public class MyReceiver  extends  BroadcastReceiver {
   public void onReceive(context,intent){}
}
```

Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyContentProvider extends  ContentProvider {
   public void onCreate(){}
}
```

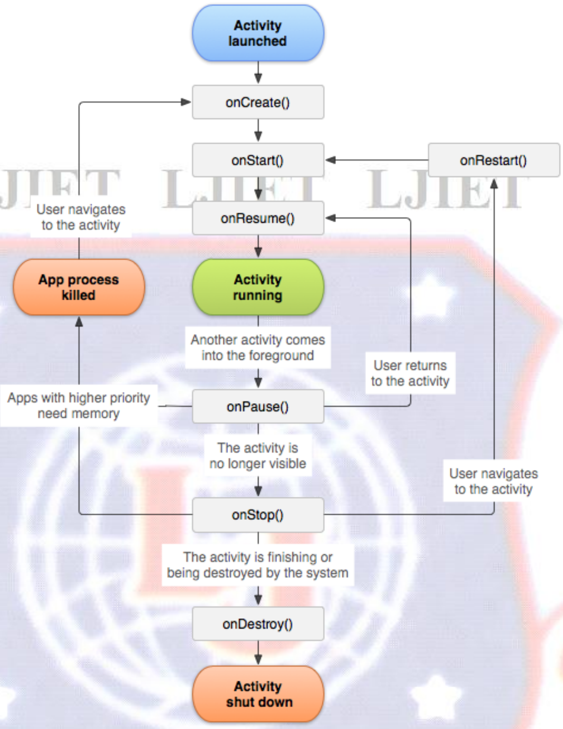
3 What is Activity Life Cycle? Explain in detail.

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

An activity is the single screen in android. It is like window or frame of Java.

By the help of activity, you can place all your UI components or widgets in a single screen.

The 7 lifecycle method of Activity describes how activity will behave at different states.



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

onCreate()
On activity creation, the activity enters the Created state. In the onCreate() method, you perform basic application startup logic that should happen only once for the entire life of the activity.

onStart()
When the activity enters the Started state, the system invokes this callback. The onStart() call makes the activity visible to the user, as the app prepares for the activity to enter the foreground and become interactive. For example, this method is where the app initializes the code that maintains the UI.

onResume()
When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the onResume() callback.
This is the state in which the app interacts with the user. The app stays in this state until something happens to take focus away from the app.
Such an event might be, for instance, receiving a phone call, the user's navigating to another activity, or the device screen's turning off.

onPause()
The system calls this method as the first indication that the user is leaving your activity (though it does not always mean the activity is being destroyed);
it indicates that the activity is no longer in the foreground (though it may still be visible if the user

is in multi-window mode).
Use the onPause() method to pause or adjust operations that should not continue (or should continue in moderation) while the Activity is in the Paused state, and that you expect to resume shortly.

onStop()

When your activity is no longer visible to the user, it has entered the Stopped state, and the system invokes the onStop() callback.
This may occur, for example, when a newly launched activity covers the entire screen. The system may also call onStop() when the activity has finished running, and is about to be terminated.

onDestroy()

onDestroy() is called before the activity is destroyed. The system invokes this callback either because: the activity is finishing (due to the user completely dismissing the activity or due to finish() being called on the activity), or

the system is temporarily destroying the activity due to a configuration change (such as device rotation or multi-window mode)

4 What you mean by fragment in Android? Explain fragment with an example

- A Fragment represents a reusable portion of your app's UI.
- Fragments cannot live on their own--they must be hosted by an activity or another fragment.
- A Fragment is a piece of an activity which enable more modular activity design.
- Android Fragment is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity.
- Fragments represent multiple screen inside one activity.
- Android fragment lifecycle is affected by activity lifecycle because fragments are included in activity.
- Each fragment has its own life cycle methods that is affected by activity life cycle because fragments are embedded in activity

Example of Fragment

Steps

1. Create another layout xml file for fragment
2. Create fragment (By Extending the fragment class)
3. Set the layout xml file to fragment
4. Use fragment tag to include fragment in xml layout

1. Create Sample Blank fragment in xml layout:

Fragment_sample.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".sample"
    android:background="@color/fragment_color">

</LinearLayout>
```

2. Sample.java

```
package com.example.simplefragmentdemo;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class sample extends Fragment {


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_sample, container, false);
    }
}
```

3. AcivityMain.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:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@color/activity_color">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/sampleFragment"
        android:name="com.example.simplefragmentdemo.sample"
        android:layout_margin="15dp"/>

</LinearLayout>
```

4. MainActivity.java

```
package com.example.simplefragmentdemo;
```

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

5 What is intent? Explain its types with appropriate Example.

Intent

• Android Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc.
• Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also.
• Intent are used for communicating between the Application components and it also provides the connectivity between two apps.

Program for Implicit and Explicit Intent

We are going to create a Screen – 1 with EditText & 2 Buttons. By clicking on 1st button user will redirect to the entered URL. (This is Implicit Intent)

2nd button – By clicking on this button User will redirect to the second activity.

As well as Screen – 2 with Button and

1. Activity_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:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">


    <EditText
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:id="@+id/editTextData"
        android:layout_marginTop="100dp"
        android:layout_marginLeft="20dp"
        android:ems="10"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/implicit_button"
```

```
            android:text="Click Here"
            android:layout_marginTop="120dp"
            android:layout_marginLeft="50dp"
            />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/explicit_button"
            android:text="Click To Go Second"
            android:layout_marginTop="130dp"
            android:layout_marginLeft="50dp"
            />
    </LinearLayout>
```

2. Activiy_second.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:layout_height="match_parent"
        tools:context=".SecondActivity"
        android:background="@color/purple_200">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This is Second Activity"
            android:id="@+id/second_text_view"
            android:layout_marginTop="130dp"
            android:layout_marginLeft="50dp"
            />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/second_button"
            android:text="Click To Go First"
            android:layout_marginTop="130dp"
            android:layout_marginLeft="20dp"
            />
    </LinearLayout>
```

3. MainActivity.java

```
package com.example.implicitexplicitintent;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private Button implicitButton, explicitButton;
    private EditText URLtext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        URLtext = findViewById(R.id.editTextData);
        implicitButton = findViewById(R.id.implicit_button);
        explicitButton = findViewById(R.id.explicit_button);
        // code for implicit intent
        implicitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String url = URLtext.getText().toString();
                Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse(url));
                startActivity(intent);
            }
        });
        // code for explicit intent
        explicitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
SecondActivity.class);
                startActivity(intent);
            }
        });

    }
}
```

4. SecondActivity.java

```
package com.example.implicitexplicitintent;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class SecondActivity extends AppCompatActivity {

    private Button secondBtn; ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        secondBtn = findViewById(R.id.second_button);
        secondBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),
MainActivity.class);
                startActivity(intent);
            }
        });

    }
}
```

6. What are widgets available in Android? Explain any two widgets with Example.

Widgets

A widget is a small gadget or control of your android application placed on the home screen.

Widgets can be very handy as they allow you to put your favourite applications on your home screen in order to quickly access them.

There are given a lot of android widgets with simplified examples such as Button, EditText, AutoCompleteTextView, ToggleButton, DatePicker, TimePicker, ProgressBar etc.

Types of widgets

Android Button
Let's learn how to perform event handling on button click.

Custom Toast
We are able to customize the toast, such as we can display image on the toast

ToggleButton
It has two states ON/OFF.

CheckBox
Let's see the application of simple food ordering.

AlertDialog
AlertDialog displays a alert dialog containing the message with OK and Cancel buttons.

Spinner
Spinner displays the multiple options, but only one can be selected at a time.

RatingBar
RatingBar displays the rating bar.

DatePicker
Datepicker displays the datepicker dialog that can be used to pick the date.

TimePicker
TimePicker displays the timepicker dialog that can be used to pick the time.

ProgressBar
ProgressBar displays progress task.

Android Button
Android Button represents a push-button. The android.widget.Button is subclass of TextView class and CompoundButton is the subclass of Button class.

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="109dp"
    android:text="ADD"
    tools:layout_editor_absoluteX="148dp"
    tools:layout_editor_absoluteY="266dp" />
```

Method:

```
button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          //code
        }
});
```

Android Toast

Andorid Toast can be used to display information for the short period of time. A toast contains message to be displayed quickly and disappears after sometime.

The android.widget.Toast class is the subclass of java.lang.Object class.

Toast class is used to show notification for a particular interval of time. After sometime it disappears. It doesn't block the user interaction.

Ex: Toast.makeText(getApplicationContext(),"Hello Javatpoint",Toast.LENGTH_SHORT).show();

7. Explain types of Menus. Write a program that shows option menu and context menu. Define xml menu file and java file.

Types of Menu:

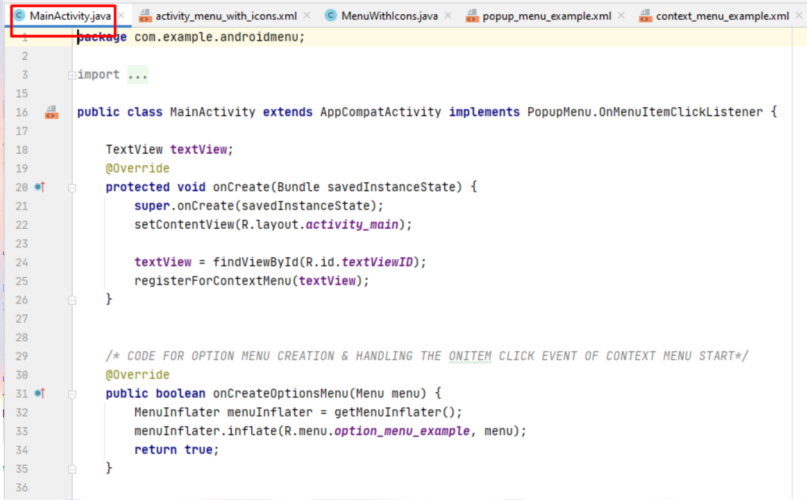
- Option Menu
- Context Menu
- Pop Up Menu

Option Menu Example

Option menu example.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/mail"
        android:title="MAIL"
        android:icon="@drawable/ic_baseline_chat_24"/>

    <item android:id="@+id/upload"
        android:title="UPLOAD"
        android:icon="@drawable/ic_baseline_contacts_24"/>

    <item android:id="@+id/share"
        android:title="SHARE"/>
</menu>
```

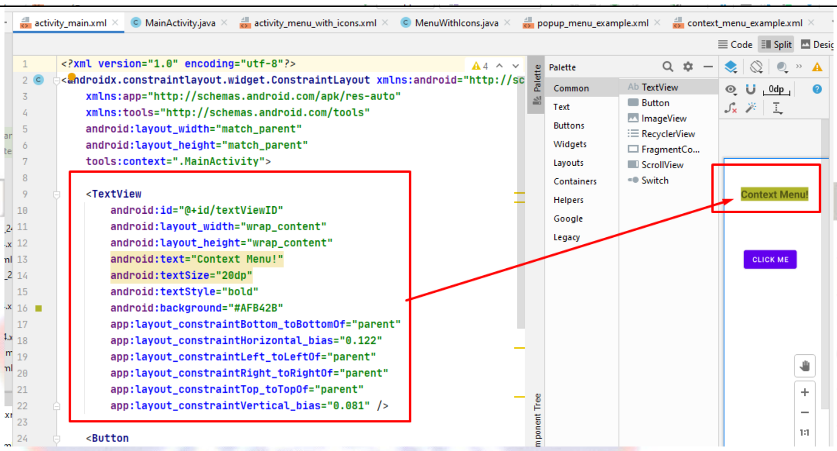
MainActivity.java File For Option Menu



```java
package com.example.androidmenu;

import ...

public class MainActivity extends AppCompatActivity implements PopupMenu.OnMenuItemClickListener {

    TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = findViewById(R.id.textViewID);
        registerForContextMenu(textView);
    }


    /* CODE FOR OPTION MENU CREATION & HANDLING THE ONITEM CLICK EVENT OF CONTEXT MENU START*/
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.option_menu_example, menu);
        return true;
    }

```

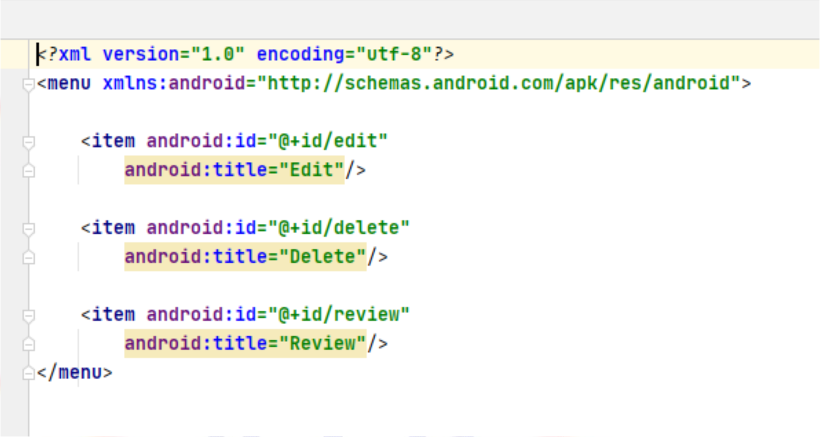
ContextMenuExample:

Activity_main.xml



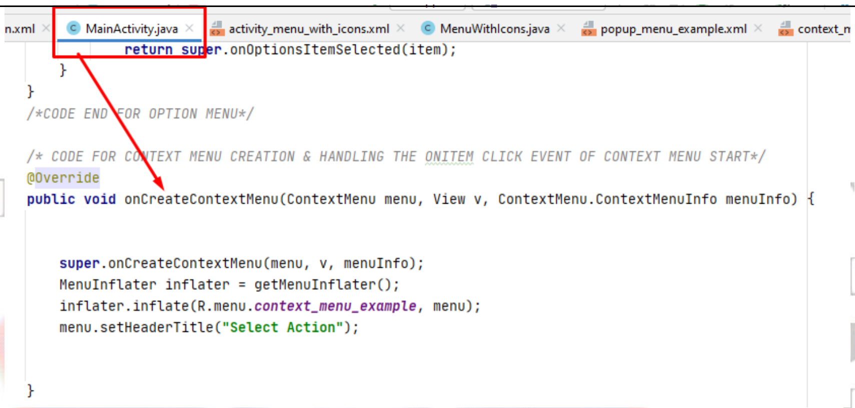
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://sc
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textViewID"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Context Menu!"
        android:textSize="20dp"
        android:textStyle="bold"
        android:background="#AFB42B"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.122"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.081" />

    <Button
```

Context_Menu_example.xml



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/edit"
        android:title="Edit"/>

    <item android:id="@+id/delete"
        android:title="Delete"/>

    <item android:id="@+id/review"
        android:title="Review"/>
</menu>
```

Just add following code in MainActivity.java file for context Menu



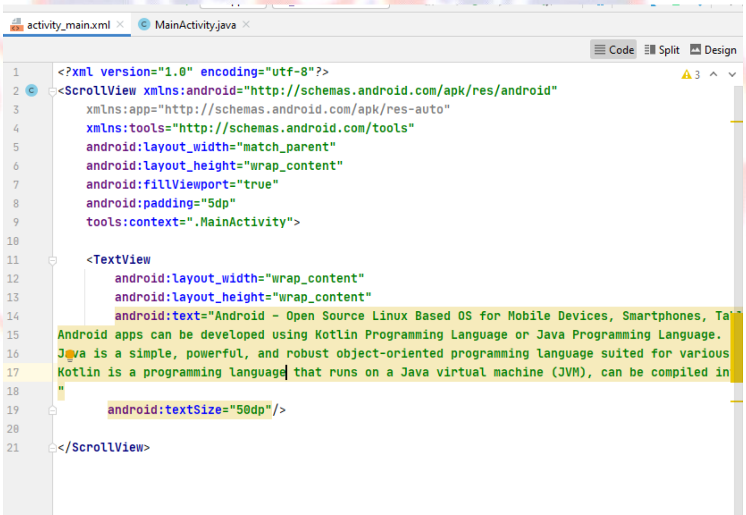
```
            return super.onOptionsItemSelected(item);
        }
    }
    /*CODE END FOR OPTION MENU*/

    /* CODE FOR CONTEXT MENU CREATION & HANDLING THE ONITEM CLICK EVENT OF CONTEXT MENU START*/
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {


        super.onCreateContextMenu(menu, v, menuInfo);
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.context_menu_example, menu);
        menu.setHeaderTitle("Select Action");


    }
```

8.

What is Scroll View and List View? Explain any one with suitable Program.

Scroll View

The android.widget.ScrollView class provides the functionality of scroll view. ScrollView is used to scroll the child elements of palette inside ScrollView.



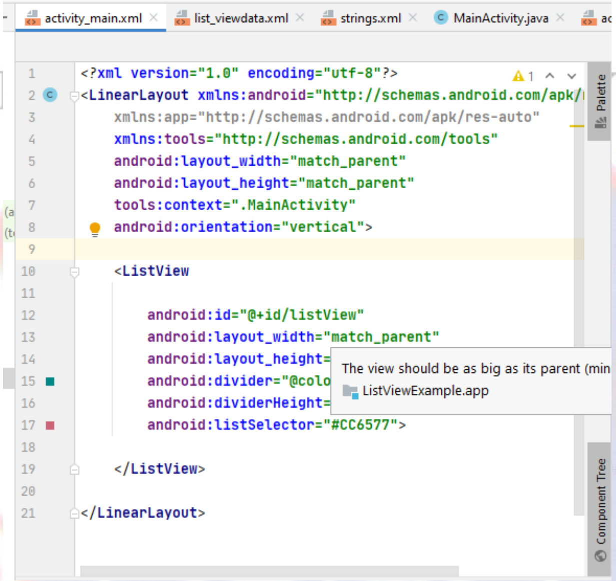
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fillViewport="true"
    android:padding="5dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Android - Open Source Linux Based OS for Mobile Devices, Smartphones, Ta
Android apps can be developed using Kotlin Programming Language or Java Programming Language.
Java is a simple, powerful, and robust object-oriented programming language suited for various
Kotlin is a programming language that runs on a Java virtual machine (JVM), can be compiled in
"
       android:textSize="50dp"/>

</ScrollView>
```

List View

- List of scrollable items can be displayed in Android using ListView.
- It helps you to displaying the data in the form of a scrollable list.
- Users can then select any list item by clicking on it.
- ListView is default scrollable so we do not need to use scroll View or anything else with ListView.
- A very common example of ListView is your phone contact book, where you have a list of your contacts displayed in a ListView and if you click on it then user information is

displayed.

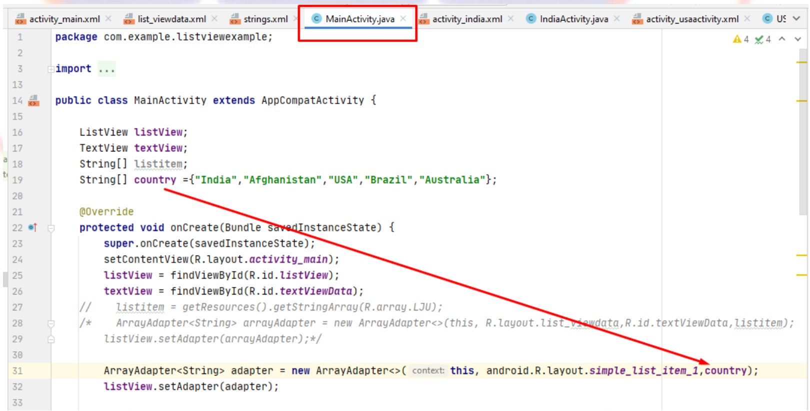
Activity_main.xml file for List View



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

    <ListView

        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height=
        android:divider="@colo
        android:dividerHeight=
        android:listSelector="#CC6577">

    </ListView>

</LinearLayout>
```

The view should be as big as its parent (min
ListViewExample.app

MainActivity.java



```
package com.example.listviewexample;

import ...

public class MainActivity extends AppCompatActivity {

    ListView listView;
    TextView textView;
    String[] listitem;
    String[] country ={"India","Afghanistan","USA","Brazil","Australia"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = findViewById(R.id.listView);
        textView = findViewById(R.id.textViewData);
//      listitem = getResources().getStringArray(R.array.LJU);
/*      ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(this, R.layout.list_viewdata,R.id.textViewData,listitem);
        listView.setAdapter(arrayAdapter);*/

        ArrayAdapter<String> adapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1,country);
        listView.setAdapter(adapter);

```

9. Write a program to insert EmployeeDetails(Eid, Ename,Esalary) in SQLite database using Android App.

DatabaseHelper.java

package com.example.sqlitedbstorageex2;

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {
 private static final int DATABASE_VERSION = 1;

   public DatabaseHelper(Context context) {
       super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
       db.execSQL("create Table Employee_Details(EID INT primary key, EName TEXT, ESalary
TEXT )");

   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

       db.execSQL("drop Table if exists Employee_Details ");
   }

   public boolean insertUserData(String ename, String salary, String eid)
   {
       SQLiteDatabase db = this.getWritableDatabase();
       ContentValues contentValues = new ContentValues();
       contentValues.put("EName",ename);
       contentValues.put("ESalary",salary);
       contentValues.put("EID",eid);
       long result = db.insert("Employee_Details ",null,contentValues);
       if(result==-1)
       {
           return false;
       }
       else
       {
           return true;
       }
   }
}
```

MainActivity.java

MainActivity.java

```
package com.example.sqlitedbstorageex2;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    //declare the variable
    Button insertBtn,
    EditText EnameEdt, ESalaryEdt, EID;
    TextView showDataTV;
    DatabaseHelper db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // map the variables
        insertBtn = findViewById(R.id.insertButton);



        db = new DatabaseHelper(this);

        insertBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String enameTxt = EnameEdt.getText().toString();
                String esalaryTxt = ESalary.getText().toString();
                String eidTxt = EID.getText().toString();


                Boolean checkInsertData = db.insertUserData(enameTxt, esalaryTxt, eidTxt);
                if(checkInsertData)
                {
                    Toast.makeText(MainActivity.this,"Insertion Succesfully
Done",Toast.LENGTH_LONG).show();
                }
                else
                {
                    Toast.makeText(MainActivity.this,"Insertion Failed",Toast.LENGTH_LONG).show();
                }
            }
        });
}
```

10. Write a program to fetch a record from table in SQLite. Where table name is mad_table and database name is mad_db. Create DatabaseHelper class file, Main Activity Java file.

DatabaseHelper.java

```
package com.example.sqlitedbstorageex2;
```

```
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {
   private static final String DATABASE_NAME= "mad_db ";
   private static final String TABLE_User= “mad_table";
   private static final String KEY_ID= "id";
   private static final String KEY_NAME= "name";
   private static final String KEY_AGE = "age";
   private static final int DATABASE_VERSION = 1;

   public DatabaseHelper(Context context) {
      super(context, DATABASE_NAME, null, DATABASE_VERSION);
   }

   @Override
   public void onCreate(SQLiteDatabase db) {
      db.execSQL("create Table mad_table (ID INT primary key, Name TEXT, age TEXT )");

   }

   @Override
   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

      db.execSQL("drop Table if exists mad_table");
   }

   public Cursor getUserData()
   {
      SQLiteDatabase db = this.getWritableDatabase();

      Cursor cursor = db.rawQuery("Select * from mad_table", null);
      return cursor;
   }

}
```

MainActivity.java File Code

```
      viewBtn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Cursor cursor = db.getUserData();
            if(cursor.getCount()==0)
            {
               Toast.makeText(MainActivity.this,"No Data Exists",Toast.LENGTH_LONG).show();
               return;
            }
            else{
               StringBuffer stringBuffer = new StringBuffer();
               while(cursor.moveToNext())
               {
                    stringBuffer.append(" Name : "+cursor.getString(0)+"\n");
                    stringBuffer.append(" Number : "+cursor.getString(1)+"\n");
                    stringBuffer.append(" Age : "+cursor.getString(2)+"\n\n");
                }

                showDataTV.setText(stringBuffer.toString());
            }
        }
    });
```

11. What are the steps to integrate retrofit in our project? Explain JSON parsing using Retrofit with program. Write code for java, model and interface files.

1. Add dependencies in you're app/build.gradle file

```
implementation 'com.squareup.retrofit2:retrofit:2.3.0'
implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
```

What are the Steps for Retrofit Data Fetching Project

1. Add Pre requisites in your project
2. Create a Simple POJO - Model Class as per your requirements of data.
3. Create an Interface (API Interface) With return Retrofit call of model class type
4. Create Retrofit Object in MainActivity.java File
5. Convert JSON Data to Model Class Object
6. Create Call of Model Class & Enqueue it For Processing
7. Receive Response Data in Simple Model Type List

Model.java

```
package com.example.retrifitex1;

public class Model {
    private int userId, id;
    private String title, body;

    public Model(int userId, int id, String title, String body) {
        this.userId = userId;
        this.id = id;
        this.title = title;
        this.body = body;
    }
    public int getUserId() {
        return userId;
    }

    public void setUserId(int userId) {
        this.userId = userId;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getBody() {
        return body;
    }

    public void setBody(String body) {
        this.body = body;
    }
}
```

Now create an interface API for fetch data.

MyAPI.interface

```
package com.example.retrifitex1;


import java.util.List;

import retrofit2.Call;
import retrofit2.http.GET;

public interface MyAPI {



    // this is call for get model data. Here posts is the file name or we can
say the URL name

    @GET("posts")
    Call<List<Model>> getModelData();

}
```

MainActivity.Java file for Retrofit object creation & fetching data (Enqueue data)

```
package com.example.retrifitex1;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;
import android.widget.Toast;

import com.google.gson.Gson;

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class MainActivity extends AppCompatActivity {


    TextView dataTv;
    String jsonURL = "https://jsonplaceholder.typicode.com/";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dataTv = findViewById(R.id.dataTextView);


        //1.. create retrofit object


        Retrofit retrofit = new Retrofit.Builder()
                .baseUrl(jsonURL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();


        // 2 .. Convert json data to model class object

        MyAPI myAPI = retrofit.create(MyAPI.class); /// json data will be
converted to MyAPI type data



        // 3.. Create a call of model class and enqueue for processing

        Call<List<Model>> call = myAPI.getModelData();

        call.enqueue(new Callback<List<Model>>() {

            @Override
            public void onResponse(Call<List<Model>> call,
Response<List<Model>> response) {
```

```
// 4. Add data to the model class object & set to the text view
                List<Model> data = response.body();

                for(int i =0; i<data.size();i++)
                {
                    dataTv.append(" Sr no: "+ data.get(i).getId() + " \n " +
data.get(i).getTitle() + "\n \n \n ");

                }
            }

            @Override
            public void onFailure(Call<List<Model>> call, Throwable t) {

            }
        });
    }
}
```

12. What is parsing? Discuss how you can perform parsing using JSON in Android Application. (Explain any one with example)

Parsing means converting json data in to the java format and java format data to json.

- JSON stands for JavaScript Object Notation.
- This an independent data exchange format and is the best alternative for XML.
- Android provides four different classes to manipulate JSON data. These classes are JSONArray,JSONObject,JSONStringer and JSONTokenizer.
- The main advantage of JSON is, it's a language independent and the JSON object will contain data like key/value pair.

Code For JSON Parsing

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;

public class MainActivity extends AppCompatActivity {


    ListView lv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        String jsondata = getListData();

        ArrayList<HashMap<String, String>> userlist = new ArrayList();
        lv = findViewById(R.id.user_list);

        try {
            // create json object
            JSONObject jsonObject = new JSONObject(jsondata);

            // get users array from the json object
            JSONArray jsonArray = jsonObject.getJSONArray("users");

            for(int i=0; i<jsonArray.length();i++)
            {
                HashMap<String,String> user = new HashMap<>();
                JSONObject obj = jsonArray.getJSONObject(i);
                user.put("name",obj.getString("name"));
                user.put("designation",obj.getString("designation"));
                user.put("location",obj.getString("location"));
                userlist.add(user);
            }

            ListAdapter listAdapter = new
SimpleAdapter(MainActivity.this,userlist,R.layout.list_row,new
String[]{"name","designation","location"},new
int[]{R.id.name,R.id.designation,R.id.location});
            lv.setAdapter(listAdapter);

        } catch (JSONException e) {
            Log.e("JsonParser Example","unexpected JSON exception", e);
        }


        // create json array
    }

    private String getListData() {
        String jsonStr = "{ \"users\" :[" +
                "{\"name\":\"Jenis Shah\",\"designation\":\"Asst.
Prof\",\"location\":\"LJIET\"}" +
                ",{\"name\":\"Ramesh Shah\",\"designation\":\"Placement
Officer\",\"location\":\"LJMCA\"}" +
                ",{\"name\":\"Mahesh Shah\",\"designation\":\"
Accountant\",\"location\":\"LJCOM\"}] }";
        return jsonStr;
    }
}
```

13. Explain the internal storage with suitable example.

Internal storage

In android, Internal Storage is useful to store the data files locally on the device’s internal memory using a FileOutputStream object. After storing the data files in device internal storage, we can read the data file from the device using a FileInputStream object.

The data files saved in the internal are managed by an android framework and it can be accessed anywhere within the app to read or write data into the file, but it’s not possible to access the file from any other app so it’s secured. When the user uninstalls the app, automatically these data files will be removed from the device internal storage.

Permission can be given as below:

NO Permission is required for Manifest File

Activity For Internal Storage.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:layout_height="match_parent"
    android:orientation="vertical "
    tools:context=".InternalStorageEx"
    android:background="#C4F4FA">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap content"
        android:id="@+id/nameEdt "
        android:layout_margin="20dp"
        android:hint="Enter YourName"
        android:inputType="text"
        android:gravity="fill_horizontal"
        />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap content"
        android:id="@+id/ageEdt "
        android:layout_margin="20dp"
        android:hint="Enter Your Age"
        android:inputType="text"
        android:gravity="fill_horizontal"
        />


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Write"
        android:id="@+id/internalStorageWriteDataBtn"
        android:layout_marginStart="20dp"/>

</LinearLayout>
```

InternalStorage.Java

```
package com.example.androidstorageex;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class InternalStorageEx extends AppCompatActivity {

    EditText ageEdt,nameEdt;
    Button writeData;
    String filename = "MSE_2_LJ.txt";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_internal_storage_ex);

        nameEdt = findViewById(R.id.nameEdt);
       ageEdt = findViewById(R.id.ageEdt);
        writeData = findViewById(R.id.internalStorageWriteDataBtn);
        writeData.setOnClickListener(new View.OnClickListener() {
          @Override
            public void onClick(View v) {
               try{
                    FileOutputStream fos =
openFileOutput(filename,MODE_PRIVATE);
                    fos.write("name =
"+nameEdt.getText().toString().getBytes());
fos.write("\n Age = "+ageEdt.getText().toString().getBytes());
                    fos.close();
                }catch (FileNotFoundException e){
                    e.printStackTrace();
                }
                catch (IOException e)
                {
                    e.printStackTrace();
                }
            }
        });



    }
}
```

14.	Write 4 methods of Text To Speech in android and explain it with suitable program. **Methods of Text To Speech**

- int speak (String text, int queueMode, HashMap params)
- int setSpeechRate(float speed)
- int setLanguage (Locale loc)
- void shutdown()
- int stop()

Activity_Main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Text To Speech Ex"
        android:layout_centerHorizontal="true"
        android:textSize="20dp"
        android:layout_margin="20dp"
        android:id="@+id/titleTextView"
         />


    <EditText
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:ems="15"
        android:id="@+id/textEdt"
        android:layout_below="@id/titleTextView"
        android:layout_margin="20dp"
        android:textColor="#AAFFDD"
        android:textColorHint="#AADDAA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textToSpeechBtn"
        android:text="Text To Speech"
        android:layout_below="@+id/textEdt"
        android:layout_centerHorizontal="true"
        android:layout_margin="20dp"
        />

</RelativeLayout>
```

MainActivity.java

```java
package com.example.texttospeechex1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.Locale;
```

```
public class MainActivity extends AppCompatActivity {

    EditText textEdt;
    Button ttsBtn;
    TextToSpeech tts;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textEdt = findViewById(R.id.textEdt);
        ttsBtn = findViewById(R.id.textToSpeechBtn);

        tts = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status!= TextToSpeech.ERROR)
                {
                    tts.setLanguage(Locale.UK);
                }
            }
        });

        ttsBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String toSpeak = textEdt.getText().toString();
                Toast.makeText(MainActivity.this, "Data = "+toSpeak,
Toast.LENGTH_SHORT).show();
                tts.speak(toSpeak,TextToSpeech.QUEUE_FLUSH,null);
            }
        });
    }

    @Override
    protected void onPause() {
        super.onPause();
        if(tts!=null)
        {
            tts.stop();
            tts.shutdown();
        }
    }
}
```

15. What are the types of permissions? Write a 2 liner code to ask permission in android. Draw flow for Run time permission in android and explain it. Write permission in manifest file for get location and external file storage.

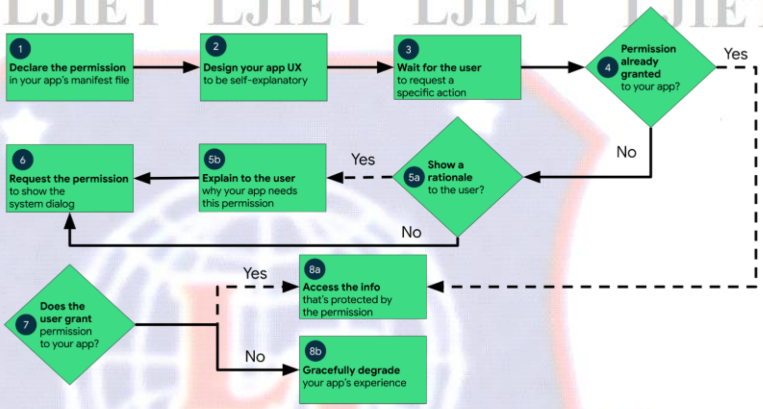
Types of Permissions

Run time permissions & Install time permissions

2 Liner Code to ask Permission in Android

```
ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},STORAGE_PERMISSION_CODE);
```

Flow Diagram for Run time Permission



1. In your app's manifest file, declare the permissions that your app might need to request.
2. Design your app's UX so that specific actions in your app are associated with specific runtime permission.
3. Wait for the user to invoke the task or action in your app that requires access to specific private user data. At that time, your app can request the runtime permission that's required for accessing that data.
4. Check whether the user has already granted the runtime permission that your app requires. If so, your app can access the private user data. If not, continue to the next step.
5. Check whether your app should show a rationale to the user, explaining why your app needs the user to grant a particular runtime permission. If the system determines that your app shouldn't show a rationale, continue to the next step directly, without showing a UI element.
6. Request the runtime permission that your app requires in order to access the private user data. The system displays a runtime permission prompt, such as the one shown on the permissions overview page.
7. Check the user's response, whether they chose to grant or deny the runtime permission.
8. If the user granted the permission to your app, you can access the private user data. If the user denied the permission instead, gracefully degrade your app experience so that it provides functionality to the user, even without the information that's protected by that permission.

Permission For Location

```
<!-- Always include this permission -->
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

  <!-- Include only if your app benefits from precise location access. -->
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Permission For External Storage is as below

```
 <uses-permission android:name =
"android.permission.WRITE_EXTERNAL_STORAGE"/>

   <uses-permission android:name =
"android.permission.READ_EXTERNAL_STORAGE"/>
```

16. Write a java code to get current location on onClick event of a get location button. Also design XML file for that.

XML File For Get Current Location

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/myMap"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" />
```

Java Code For Fetch Current Location

```
 private void fetchLocation() {
      if (ActivityCompat.checkSelfPermission(
         this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(
         this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
         ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_CODE);
         return;
      }
      Task<Location> task =
fusedLocationProviderClient.getLastLocation();
      task.addOnSuccessListener(new OnSuccessListener<Location>() {
         @Override
         public void onSuccess(Location location) {
            if (location != null) {
               currentLocation = location;
```

```
                Toast.makeText(getApplicationContext(),
currentLocation.getLatitude() + "" + currentLocation.getLongitude(),
Toast.LENGTH_SHORT).show();
                SupportMapFragment supportMapFragment =
(SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.myMap);

                supportMapFragment.getMapAsync(MainActivity.this);
            }
        }
    });
}
```

17. What are the types of animations in android? Write a program to perform Tween Animation (Zoom, Anti clock wise rotation) with design, xml file of rotation and Java file.

Types of Animation in Android

- Bitmap Animation
- Frame Animation
- Tween Animation
- View Animation

Rotate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:andorid="http://schemas.android.com/apk/res-auto"
    android:fillAfter="true">


    <rotate xmlns:android="http://schemas.android.com/apk/res/android"
        android:startOffset="5000"
        android:fromDegrees="360"
        android:toDegrees="0"
        android:pivotX="70%"
        android:pivotY="50%"
        android:duration="2500" >
    </rotate>

</set>
```

Zoom.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromXScale="0.5"
        android:toXScale="3.0"
        android:fromYScale="0.5"
        android:toYScale="3.0"
        android:duration="4000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>


    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:startOffset="5000"
        android:fromXScale="3.0"
        android:toXScale="0.5"
        android:fromYScale="3.0"
        android:toYScale="0.5"
        android:duration="4000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>

</set>
```

Java Code for both animations

```
rotateBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.rotate_animation);
        logo.startAnimation(animation);
    }
});
zoomBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom_animation);
        objectAnimator =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom_animation);
        logo.startAnimation(objectAnimator);
    }
});
```

18. What do we mean by Shared Preference? Write a code to skip login and directly move towards Home Screen Activity with the help of Shared preference. Write JAVA files only.

In android, Shared Preferences are used to save and retrieve the primitive data types (integer, float, boolean, string, long) data in the form of key-value pairs from a file within an apps file structure.

Generally, the Shared Preferences object will point to a file that contains key-value pairs and provides a simple read and write methods to save and retrieve the key-value pairs from a file.

MainActivity.java

```
package com.example.sharedpreferenceexample;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                SharedPreferences sharedPreferences =
getSharedPreferences("login",MODE_PRIVATE); // create the sharedpreference
                Boolean loginCheck =
sharedPreferences.getBoolean("flag",false); //


                Intent intent;

                if(loginCheck)
                {
                    intent = new Intent(MainActivity.this,HomeScreen.class);
                }
                else{

                    intent = new Intent(MainActivity.this,LoginActivity.class);
                }
                startActivity(intent);
            }
        },4000);
    }
}
```

HomeScreen.java

```
package com.example.sharedpreferenceexample;

import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class HomeScreen extends AppCompatActivity {

    Button logoutButton;
    TextView homeScreenTV;
    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home_screen);

        logoutButton = findViewById(R.id.logoutButton);
        homeScreenTV = findViewById(R.id.homeScreenTV);

        logoutButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                SharedPreferences sharedPreferences =
getSharedPreferences("login",MODE_PRIVATE);
                SharedPreferences.Editor editor = sharedPreferences.edit();

                String name = sharedPreferences.getString("Name",null);

                homeScreenTV.setText(name);
                editor.putBoolean("flag",false);
                editor.apply();
            }
        });
    }
}
```

You can check shared preference in Device Explorer window.

Make sure – You have to run app and click on Login button then and only then u can check it

19. Write a code to enter name, age and number from Activity and store in external storage files.

Write program for all necessary files.

Activity For Internal Storage.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:layout_height="match_parent"
    android:orientation="vertical "
    tools:context=".InternalStorageEx"
    android:background="#C4F4FA">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap content"
        android:id="@+id/nameEdt "
        android:layout_margin="20dp"
        android:hint="Enter YourName"
        android:inputType="text"
        android:gravity="fill_horizontal"
        />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap content"
        android:id="@+id/ageEdt "
        android:layout_margin="20dp"
        android:hint="Enter Your Age"
        android:inputType="text"
        android:gravity="fill_horizontal"
        />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/numEdt "
        android:layout_margin="20dp"
        android:hint="Enter Your Number"
        android:inputType="text"
        android:gravity="fill_horizontal"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Write"
        android:id="@+id/externalStorageWriteDataBtn"
        android:layout_marginStart="20dp"/>

</LinearLayout>
```

InternalStorage.Java

```
package com.example.androidstorageex;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

public class InternalStorageEx extends AppCompatActivity {

    EditText ageEdt,nameEdt, numEdt;
    Button writeData;

        String filename = "myExternalStorage_Live.txt";
       String filepath = "myExternalDirectory_Live";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_internal_storage_ex);

        nameEdt = findViewById(R.id.nameEdt);
       ageEdt = findViewById(R.id.ageEdt);
      numEdt = findViewById(R.id.numEdt);
        writeData = findViewById(R.id.externalStorageWriteDataBtn);
filepath = "myExternalDirectory_Live";
        if(!isExternalStorageAvailableForRW())
        {
            writeData.setEnabled(false);
        }
        writeData.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showData.setText("");
```

```
                 filecontent =nameEdt.getText().toString()+”\n”+
ageEdt.getText().toString()+numEdt.getText().toString();
                 if(!filecontent.equals(""))
                 {
                     File myExternalFile = new
File(getExternalFilesDir(filepath),filename);
                     FileOutputStream fos = null;
                     try {
                         fos = new FileOutputStream(myExternalFile);
                         fos.write(filecontent.getBytes());
                     } catch (FileNotFoundException e) {
                         e.printStackTrace();
                     } catch (IOException e) {
                         e.printStackTrace();
                     }
                     externalDataEdt.setText("");
                     Toast.makeText(MainActivity.this, "Information Saved
Succesfully", Toast.LENGTH_SHORT).show();

                 }
                 else {
                     Toast.makeText(MainActivity.this,"Text FIled Can not be
empty",Toast.LENGTH_SHORT).show();
                 }
             }
         });
```

20. What do we mean by web view? Write a program with design and java file to integrate web view in android.

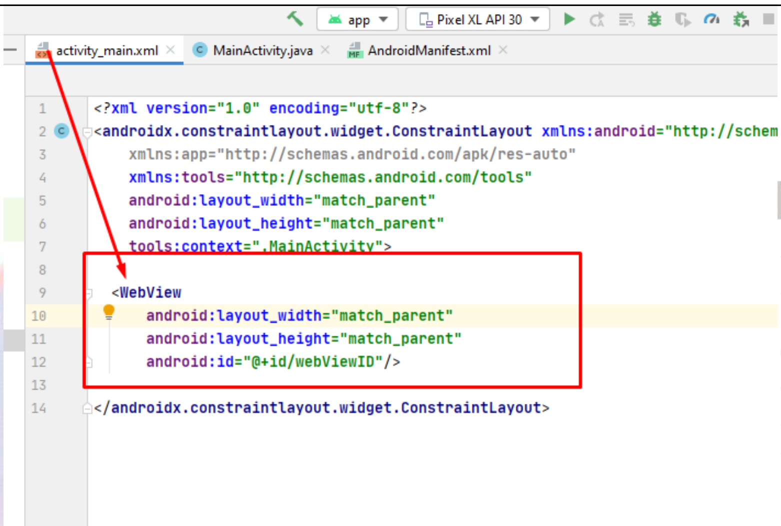
WebView

Android WebView is used to display web page in android. The web page can be loaded from same application or URL. It is used to display online content in android activity.

Program for Web View

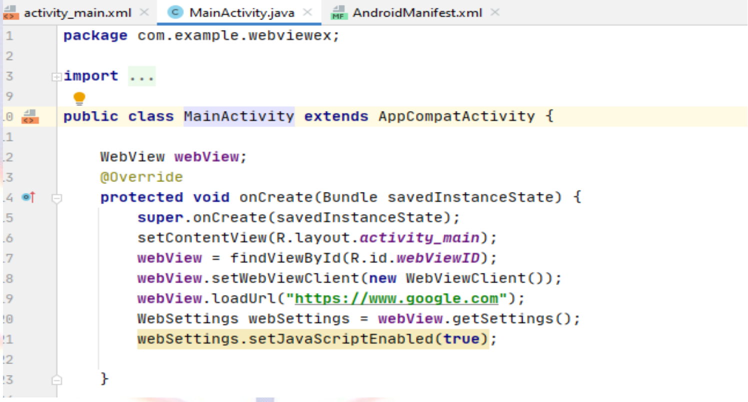
```
<!--Add this before application tag in AndroidManifest.xml-->
<uses-permission android:name="android.permission.INTERNET" />
```

Activity_main.xml



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schem
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

  <WebView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/webViewID"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java



```java
package com.example.webviewex;

import ...

public class MainActivity extends AppCompatActivity {

    WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        webView = findViewById(R.id.webViewID);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("https://www.google.com");
        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);

    }
```

21. What is service in android? Draw and explain service life cycle. Write a program that shows example of service in android.

Service in Android

- Android service is a component that is used to perform operations on the background such as playing music, handle network transactions, interacting content providers etc.
- It doesn't has any UI (user interface).

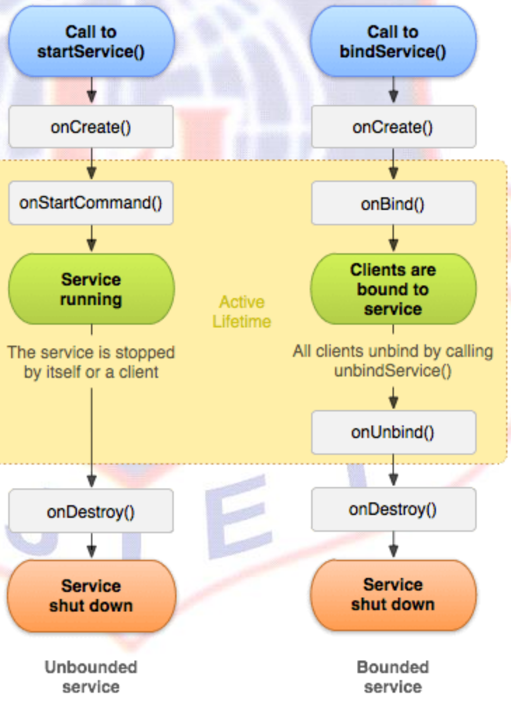
Life Cycle of Android Service

There can be two forms of a service.The lifecycle of service can follow two different paths: started or bound.

Started
Bound
1) Started Service
A service is started when component (like activity) calls startService() method, now it runs in the background indefinitely. It is stopped by stopService() method. The service can stop itself by calling the stopSelf() method.

2) Bound Service
A service is bound when another component (e.g. client) calls bindService() method. The client can unbind the service by calling the unbindService() method.

Below is the Service Life Cycle



Program For Service

Add Following code in Manifest File.xml

```
<service android:name=".MyService"/>
```

Activity_main.xml file for Start & Stop Service

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/nameTV"
        android:text="SERVICE EXAMPLE"
        android:textSize="30dp"
        android:layout_margin="50dp"
        android:layout_centerHorizontal="true"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="START"
        android:id="@+id/startBtn"
        android:textSize="10dp"
        android:layout_below="@+id/nameTV"
        android:layout_margin="20dp"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="STOP"
        android:id="@+id/stopBtn"
        android:textSize="10dp"
        android:layout_below="@+id/nameTV"
        android:layout_margin="20dp"
        android:layout_alignParentEnd="true"/>
</RelativeLayout>
```

My Service.java File (This is the Service class that extends Service)

```
package com.example.serviceexample1;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.media.audiofx.Equalizer;
import android.os.IBinder;
import android.provider.Settings;
import android.widget.MediaController;
import android.widget.Toast;

import androidx.annotation.Nullable;

public class MyService extends Service {


    private MediaPlayer player;
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {


        player = MediaPlayer.create(this,
Settings.System.DEFAULT_ALARM_ALERT_URI);

        player.setLooping(true);

        player.start();

        Toast.makeText(this,"Service Started",Toast.LENGTH_LONG).show();
        return START_STICKY;
    }


    @Override
    public void onDestroy() {
        super.onDestroy();
        player.stop();
        Toast.makeText(this,"Service Destroyed",Toast.LENGTH_LONG).show();

    }

}
```

MainActivity.java

```
package com.example.serviceexample1;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    String msg = "Android";
    Button startBtn, stopBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(msg,"OnCreate Method Called");

        startBtn = findViewById(R.id.startBtn);
        stopBtn = findViewById(R.id.stopBtn);
        startBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                startService(new Intent(getBaseContext(),MyService.class));
            }
        });

        stopBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                stopService(new Intent(getBaseContext(),MyService.class));
                }
            });
        }
}
```

22. Write a program to show the paired devices by Bluetooth. Write all necessary files of it.

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.bluetoothex1">

    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.BluetoothEx1">
        <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>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Bluetooth Example"
        android:id="@+id/titleTextView"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp"
        android:textSize="20dp"

       />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/listDeviceBtn"
        android:layout_centerHorizontal="true"
        android:text="Paired Devices List"
        android:id="@+id/pairedTextView"
        android:textSize="30dp"
        android:textStyle="bold|italic"
        />
    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listOfDevices"
        android:layout_below="@+id/pairedTextView"
        android:divider="@color/black"
        android:dividerHeight="2dp"
        android:layout_margin="30dp"
        />

        />

</RelativeLayout>
```

MainActivity.java

```
package com.example.bluetoothex1;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;

public class MainActivity extends AppCompatActivity {

    Button pairedDevicesBtn;
    ListView listOfPairedDevices;
    private BluetoothAdapter BA;
    private Set<BluetoothDevice> pairedDevices;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        pairedDevicesBtn = findViewById(R.id.listDeviceBtn);
        listOfPairedDevices = findViewById(R.id.listOfDevices);
        BA = BluetoothAdapter.getDefaultAdapter();

        pairedDevicesBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                pairedDevices = BA.getBondedDevices();
                ArrayList list = new ArrayList();
                for(BluetoothDevice bt : pairedDevices)
                {
                    list.add(bt.getName());
                }
                Toast.makeText(MainActivity.this, "Showing Paired Devices",
Toast.LENGTH_SHORT).show();

                final ArrayAdapter adapter = new
ArrayAdapter(getApplicationContext(),
android.R.layout.simple_list_item_1,list);

                listOfPairedDevices.setAdapter(adapter);
            }
        });


    }
}
```

23. How to take pictures with camera on Android Programmatically? Write code for all necessary files.

In Manifest file : No Extra permissions are needed

Acitivity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Camera Ex"
        android:id="@+id/titleTV"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cameraBtn"
        android:layout_centerHorizontal="true"
        android:layout_margin="30dp"
        android:text="Take Photo"
        />

    <ImageView
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_below="@+id/cameraBtn"
        android:id="@+id/imageView1"
        android:layout_centerHorizontal="true"
        android:src="@drawable/ic_launcher_foreground"
        />
</RelativeLayout>
```

MainActivity.java

```
package com.example.cameraex1;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    private static final int CAMERA_REQUEST = 1008;
    ImageView imgview;
    Button cameraBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        cameraBtn = findViewById(R.id.cameraBtn);
        imgview = findViewById(R.id.imageView1);

        cameraBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent cameraIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent,CAMERA_REQUEST);
            }
        });


    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {

        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == CAMERA_REQUEST) {
            Bitmap photo = (Bitmap) data.getExtras().get("data");
            imgview.setImageBitmap(photo);

        }
    }
}
```

24. Write a code for example demonstrating the use of WIFI. It creates a basic application that open your wifi and close your wifi

ManifestFile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.wifiexample">

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE/"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.WifiExample">
        <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>
```

activityMain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Wifi Manager Example"
        android:layout_centerHorizontal="true"
        android:textSize="20dp"
        android:layout_margin="30dp"
        android:id="@+id/titleTextView"
         />

    <Button
        android:id="@+id/onWifiBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Enable Wifi"
        android:layout_margin="20dp"
        android:layout_below="@+id/titleTextView" />

    <Button
        android:id="@+id/offWifiBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Enable Wifi"
        android:layout_margin="20dp"
        android:layout_below="@+id/onWifiBtn" />
</RelativeLayout>
```

25. Write a program to integrate audio and video in android.

AndroidManifestFile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.androidmediaplayersimpleexamples">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AndroidMediaPlayerSimpleExamples">
        <activity android:name=".VideoPlayerEx">

            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity">

        </activity>
    </application>

</manifest>
```

Activity_main.xml (For Audio)

```
<?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:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Simple Media - Audio Player"
        android:layout_gravity="center"
        android:textSize="20dp"
        android:gravity="center"
        android:textStyle="bold"
        android:layout_margin="20dp"
         />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listenBtn"
        android:layout_gravity="center"
        android:text="Listen"
        />


</LinearLayout>
```

MainActivity.java (For Audio)

```
package com.example.androidmediaplayersimpleexamples;
```

```
import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    Button listenBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listenBtn = findViewById(R.id.listenBtn);
        listenBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                MediaPlayer mediaPlayer =
MediaPlayer.create(getApplicationContext(),R.raw.test_song);
                mediaPlayer.start();
                mediaPlayer.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {
                    @Override
                    public void onCompletion(MediaPlayer mp) {
                        Toast.makeText(getApplicationContext(),"Completed
Song",Toast.LENGTH_LONG).show();
                    }
                });
            }
        });
    }
}
```

ActivityforVideoPlayer.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:layout_height="match_parent"
    tools:context=".VideoPlayerEx">

    <VideoView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/videoViewID"
        android:layout_margin="20dp"
        />


</LinearLayout>
```

VideoPlayerEx.java (For Video)

```
package com.example.androidmediaplayersimpleexamples;

import androidx.appcompat.app.AppCompatActivity;
import android.net.Uri;
import android.os.Bundle;
import android.widget.MediaController;
import android.widget.VideoView;

public class VideoPlayerEx extends AppCompatActivity {

    VideoView videoView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video_player_ex);
        videoView = findViewById(R.id.videoViewID);
        String videoPath  = "android.resource://" +getPackageName()+ "/"
+R.raw.lj_intro;

        Uri uri = Uri.parse(videoPath);
        videoView.setVideoURI(uri);
        MediaController mediaController = new MediaController(this);
        videoView.setMediaController(mediaController);

        mediaController.setAnchorView(videoView);


    }
}
```