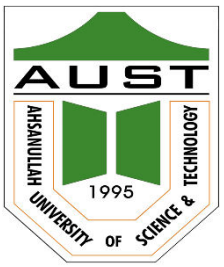


Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

LABORATORY MANUAL

Course No. : CSE2200
Course Title: Software Development III

For the students of 2nd Year, 2nd Semester of
B.Sc. in Computer Science and Engineering program

TABLE OF CONTENTS

COURSE OBJECTIVES

- Students should learn to develop applications for android gazettes.
- Students should be able to develop android applications using Android Studio.
- Students should be able to store information.

PREFFERED TOOL(S)

- Android Studio

REFERENCES

Books

1. Android Programming for Beginners (1st Ed)
   Authored by: John Horton
2. Hello Android: Introducing Google's Mobile Development Platform (1st Ed)
   Authored by: Ed Burnette
3. Head First Android Development: A Brain-Friendly Guide (1st Ed)
   Authored by: Sawn Griffiths

Online Resources

1. https://developer.android.com/index.html
2. https://developer.android.com/guide/index.html
3. https://www.tutorialspoint.com/android/index.html
4. https://developers.google.com/training/android/

ADMINISTRATIVE POLICY OF THE LABORATORY

- ✓ Students must perform class assessment tasks individually without help of others.
- ✓ Viva for each program will be taken and considered as a performance.
- ✓ Plagiarism is strictly forbidden and will be dealt with punishment.

Session 01

Goals:

1. To know about Android operating system
2. To Know about Android Studio
3. To know about some layouts in android

Android operating system:

Android is an open source operating system, based on Linux kernel and used in mobile devices like smartphones, tablets etc. Further, it was developed for smart watches and Android TV. Each of them has a specialized interface. Because it is Android Open Source Project(AOSP) licensed under the Apache license, many developers has been contributed in android development. The primary goal of the android project is to create product that can be implemented in user's life.

Android Studio:

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. The website for Android Studio is https://developer.android.com/.

Some layouts in Android:

1. Linear Layout
2. Relative Layout
3. Table Layout
4. Absolute Layout
5. Frame Layout

Linear Layout:

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally.

MainActivity.java file

```
import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
  }
}
```

activity_main.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical" >

  <Button android:id="@+id/btnStartService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="start_service"/>

  <Button android:id="@+id/btnPauseService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="pause_service"/>

  <Button android:id="@+id/btnStopService"
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:text="stop_service"/>

</LinearLayout>
```

Relative Layout:

RelativeLayout is a view group that displays child views in relative positions.

MainActivity.java file

```
import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

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

}
```

activity_main.xml file

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:paddingLeft="16dp"
  android:paddingRight="16dp" >

  <EditText
    android:id="@+id/name"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="@string/reminder" />

  <LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/name">

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="New Button"
      android:id="@+id/button" />

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="New Button"
      android:id="@+id/button2" />

  </LinearLayout>

</RelativeLayout>
```

Table Layout:
TableLayout is a view that groups views into rows and columns.

MainActivity.java file

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }

}
```

activity_main.xml file

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
      android:text="Time"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_column="1" />

    <TextClock
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/textClock"
      android:layout_column="2" />
```

```
</TableRow>

<TableRow>

  <TextView
    android:text="First Name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1" />

  <EditText
    android:width="200px"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</TableRow>

<TableRow>

  <TextView
    android:text="Last Name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="1" />

  <EditText
    android:width="100px"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</TableRow>

<TableRow
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <RatingBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/ratingBar"
    android:layout_column="2" />
</TableRow>
```

```
  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>

  <TableRow
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Submit"
      android:id="@+id/button"
      android:layout_column="2" />
  </TableRow>

</TableLayout>
```

Absolute Layout:
AbsoluteLayout enables you to specify the exact location of its children.

MainActivity.java file

```
import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }

}
```

activity_main.xml file

```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
```

```
  android:layout_height="fill_parent">

  <Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="OK"
    android:layout_x="50px"
    android:layout_y="361px" />
  <Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:text="Cancel"
    android:layout_x="225px"
    android:layout_y="361px" />

</AbsoluteLayout>
```

Frame Layout:
The FrameLayout is a placeholder on screen that you can use to display a single view.

MainActivity.java file

```
import android.os.Bundle;
import android.app.Activity;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
}
```

activity_main.xml file

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">

  <ImageView
```

```
    android:src="@drawable/ic_launcher"
    android:scaleType="fitCenter"
    android:layout_height="250px"
    android:layout_width="250px"/>

  <TextView
    android:text="Frame Demo"
    android:textSize="30px"
    android:textStyle="bold"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent"
    android:gravity="center"/>
</FrameLayout>
```

Session 02

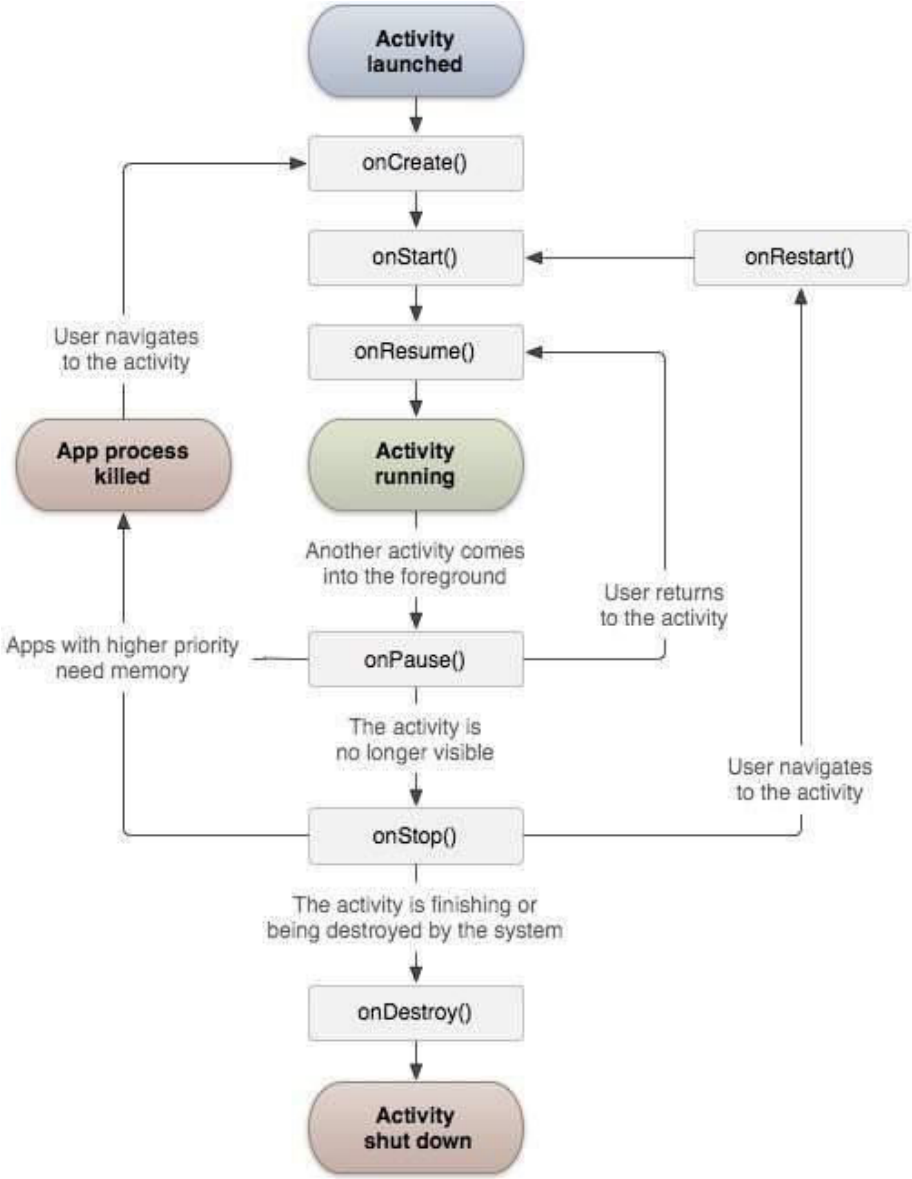
Goal(s)

To know about activity and architecture of Android OS

Activity

An Android activity is one screen of the Android app's user interface. In that way an Android activity is very similar to windows in a desktop application. An Android app may contain one or more activities, meaning one or more screens. The Android app starts by showing the main activity, and from there the app may make it possible to open additional activities.

Activity life cycle

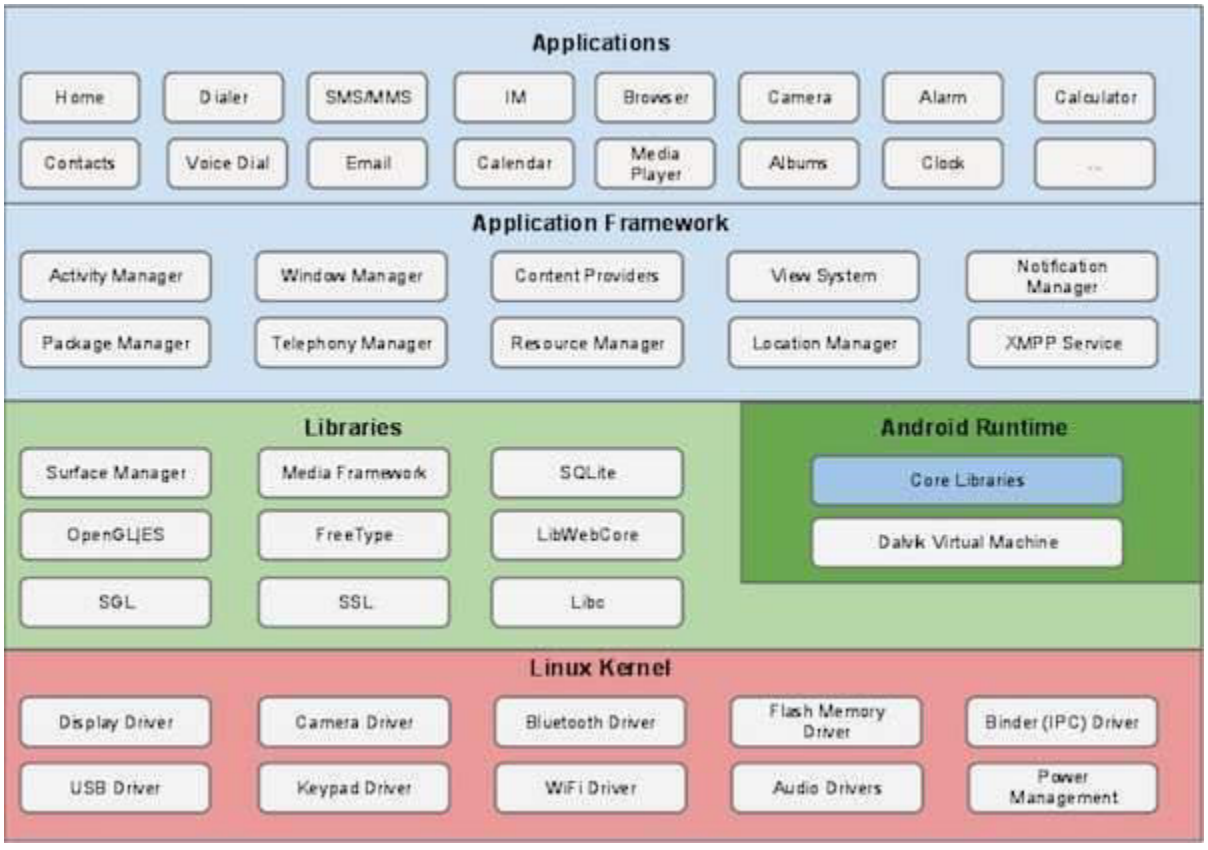


```
import android.os.Bundle;
import android.app.Activity;
import android.util.Log;

public class MainActivity extends Activity {
  String msg = "Android : ";

  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d(msg, "The onCreate() event");
  }

  /** Called when the activity is about to become visible. */
  @Override
  protected void onStart() {
    super.onStart();
    Log.d(msg, "The onStart() event");
  }

  /** Called when the activity has become visible. */
  @Override
  protected void onResume() {
    super.onResume();
    Log.d(msg, "The onResume() event");
  }

  /** Called when another activity is taking focus. */
  @Override
  protected void onPause() {
    super.onPause();
    Log.d(msg, "The onPause() event");
  }

  /** Called when the activity is no longer visible. */
  @Override
  protected void onStop() {
    super.onStop();
```

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

  /** Called just before the activity is destroyed. */
  @Override
  public void onDestroy() {
    super.onDestroy();
    Log.d(msg, "The onDestroy() event");
  }
}
```

Android Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware

drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows −

- android.app − Provides access to the application model and is the cornerstone of all Android applications.
- android.content − Facilitates content access, publishing and messaging between applications and application components.
- android.database − Used to access data published by content providers and includes SQLite database management classes.
- android.opengl − A Java interface to the OpenGL ES 3D graphics rendering API.
- android.os − Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- android.text − Used to render and manipulate text on a device display.
- android.view − The fundamental building blocks of application user interfaces.
- android.widget − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- android.webkit − A set of classes intended to allow web-browsing capabilities to be built into applications.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services −

- Activity Manager − Controls all aspects of the application lifecycle and activity stack.
- Content Providers − Allows applications to publish and share data with other applications.
- Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- Notifications Manager − Allows applications to display alerts and notifications to the user.
- View System − An extensible set of views used to create application user interfaces.

Applications

Android applications are found at the top layer. Applications are written to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

Session 03

Goals

1. To know about Splash Activity
2. To know about Android Intent

Splash Activity

Splash Screen is most commonly the first startup screen which appears when application is opened. In other words, it is a simple constant screen for a fixed amount of time which is used to display the company logo, name, advertising content etc.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="#946c6c"
  android:orientation="vertical">
  <ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/pic" />
</LinearLayout>
```

activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Second activity"
    android:textSize="40dp"
    android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

HomeActivity.java

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class HomeActivity extends AppCompatActivity {

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

MainActivity.java

```
import android.content.Intent;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

   private static int SPLASH_TIME_OUT = 4000;

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

   public void splash()
   {
      new Handler().postDelayed(new Runnable() {
         public void run() {
            Intent ob = new Intent(MainActivity.this, HomeActivity.class);
            startActivity(ob);
            finish();
         }
      },SPLASH_TIME_OUT);
   }
```

}

Android Intent

Android Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc. It is generally used with startActivity() method to invoke activity, broadcast receivers etc. The dictionary meaning of intent is intention or purpose.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Intent Example"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:textSize="30dp" />

  <TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tutorials point"
    android:textColor="#ff87ff09"
    android:textSize="30dp"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true" />

  <ImageButton
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageButton"
    android:src="@drawable/abc"
    android:layout_below="@+id/textView2"
    android:layout_centerHorizontal="true" />

  <EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_below="@+id/imageButton"
    android:layout_alignRight="@+id/imageButton"
    android:layout_alignEnd="@+id/imageButton" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Browser"
    android:id="@+id/button"
    android:layout_alignTop="@+id/editText"
    android:layout_alignRight="@+id/textView1"
    android:layout_alignEnd="@+id/textView1"
    android:layout_alignLeft="@+id/imageButton"
    android:layout_alignStart="@+id/imageButton" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Phone"
    android:id="@+id/button2"
    android:layout_below="@+id/button"
    android:layout_alignLeft="@+id/button"
    android:layout_alignStart="@+id/button"
    android:layout_alignRight="@+id/textView2"
    android:layout_alignEnd="@+id/textView2" />
</RelativeLayout>
```

MainActivity.java

```
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
  Button b1,b2;

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

    b1=(Button)findViewById(R.id.button);
    b1.setOnClickListener(new View.OnClickListener() {

      public void onClick(View v) {
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
          Uri.parse("http://www.example.com"));
        startActivity(i);
      }
    });

    b2=(Button)findViewById(R.id.button2);
    b2.setOnClickListener(new View.OnClickListener() {
      public void onClick(View v) {
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
          Uri.parse("tel:9510300000"));
        startActivity(i);
      }
    });
  }
}
```

Session 04

Goal

To know about database for Android application

Database

SQLite is an open-source SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC,ODBC etc.

Sample Example

MainActivity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends Activity {
  EditText userInput;
  TextView recordsTextView;
  MyDBHandler dbHandler;
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    userInput = (EditText) findViewById(R.id.user_Input);
    recordsTextView = (TextView) findViewById(R.id.records_TextView);
    dbHandler = new MyDBHandler(this, null, null, 1);
    printDatabase();
  }
```

```
    public void printDatabase(){
        String dbString = dbHandler.databaseToString();
        recordsTextView.setText(dbString);
        userInput.setText("");
    }

    public void addButtonClicked(View view){
            Products product = new Products(userInput.getText().toString());
        dbHandler.addProduct(product);
        printDatabase();
    }
    public void deleteButtonClicked(View view){
        String inputText = userInput.getText().toString();
        dbHandler.deleteProduct(inputText);
        printDatabase();
    }
}
```

Products.java

```
public class Products {
    private int _id;
    private String _productname;
    public Products(){

    }
    public Products(String productName) {
        this._productname = productName;
    }

    public int get_id() {
```

```
        return _id;
    }
    public void set_id(int _id) {
        this._id = _id;
    }
    public String get_productname() {
        return _productname;
    }
    public void set_productname(String _productname) {
        this._productname = _productname;
    }
}
```

MyDBHandler.java

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

public class MyDBHandler extends SQLiteOpenHelper{
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "productDB.db";
    public static final String TABLE_PRODUCTS = "products";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_PRODUCTNAME = "productname";

    public MyDBHandler(Context context, String name, SQLiteDatabase.CursorFactory
factory, int version) {
        super(context, DATABASE_NAME, factory, DATABASE_VERSION);
```

```
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String query = "CREATE TABLE " + TABLE_PRODUCTS + "(" +
                COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
                COLUMN_PRODUCTNAME + " TEXT " +
                ");";
        db.execSQL(query);
    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_PRODUCTS);
        onCreate(db);
    }

    public void addProduct(Products product){
        ContentValues values = new ContentValues();
        values.put(COLUMN_PRODUCTNAME, product.get_productname());
        SQLiteDatabase db = getWritableDatabase();
        db.insert(TABLE_PRODUCTS, null, values);
        db.close();
    }
    public void deleteProduct(String productName){
        SQLiteDatabase db = getWritableDatabase();
 db.execSQL("DELETE FROM "+TABLE_PRODUCTS+" WHERE "+COLUMN_PRODUCTNAME
+ "=\"" + productName + "\";");
    }

    public String databaseToString(){
        String dbString = "";
```

```
        SQLiteDatabase db = getWritableDatabase();
        String query = "SELECT * FROM " + TABLE_PRODUCTS + " WHERE 1";
        Cursor recordSet = db.rawQuery(query, null);
        recordSet.moveToFirst();

        while (!recordSet.isAfterLast()) {
          if (recordSet.getString(recordSet.getColumnIndex("productname")) != null) {
            dbString += recordSet.getString(recordSet.getColumnIndex("productname"));
            dbString += "\n";
          }
          recordSet.moveToNext();
        }
        db.close();
        return dbString;
    }
}
```

activity_main.xml

```
<RelativeLayout 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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
     tools:context=".MainActivity">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/user_Input"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="69dp"
        android:width="300dp" />

    <Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add"
    android:id="@+id/add_Button"
    android:layout_below="@+id/user_Input"
    android:layout_alignStart="@+id/user_Input"
    android:layout_marginTop="40dp"
    android:onClick="addButtonClicked" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Delete"
    android:id="@+id/delete_Button"
    android:layout_alignTop="@+id/add_Button"
    android:layout_alignEnd="@+id/user_Input"
    android:onClick="deleteButtonClicked" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Large Text"
    android:id="@+id/records_TextView"
    android:layout_centerVertical="true"
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Session 05

Goal

To know about animation in Android

Tween Animation

Tween Animation takes some parameters such as start value , end value, size , time duration , rotation angle e.t.c and performs the required animation on that object. It can be applied to any type of object. So in order to use this , android has provided a class called Animation.

Example Implementation

MainActivity.java.

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }

  public void clockwise(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
      R.anim.myanimation);
    image.startAnimation(animation);
  }

  public void zoom(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 = AnimationUtils.loadAnimation(getApplicationContext(),
      R.anim.clockwise);
    image.startAnimation(animation1);
```

```
  }

  public void fade(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.fade);
    image.startAnimation(animation1);
  }

  public void blink(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.blink);
    image.startAnimation(animation1);
  }

  public void move(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(), R.anim.move);
    image.startAnimation(animation1);
  }

  public void slide(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(), R.anim.slide);
    image.startAnimation(animation1);
  }
}
```

res/layout/activity_main.xml

```
<RelativeLayout
  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"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

  <TextView
```

```
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Alert Dialog"
   android:id="@+id/textView"
   android:textSize="35dp"
   android:layout_alignParentTop="true"
   android:layout_centerHorizontal="true" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Tutorialspoint"
   android:id="@+id/textView2"
   android:textColor="#ff3eff0f"
   android:textSize="35dp"
   android:layout_below="@+id/textView"
   android:layout_centerHorizontal="true" />

<ImageView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:id="@+id/imageView"
   android:src="@drawable/abc"
   android:layout_below="@+id/textView2"
   android:layout_alignRight="@+id/textView2"
   android:layout_alignEnd="@+id/textView2"
   android:layout_alignLeft="@+id/textView"
   android:layout_alignStart="@+id/textView"/>

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="zoom"
   android:id="@+id/button"
   android:layout_below="@+id/imageView"
   android:layout_alignParentLeft="true"
   android:layout_alignParentStart="true"
   android:layout_marginTop="40dp"
   android:onClick="clockwise"/>

<Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="clockwise"
   android:id="@+id/button2"
   android:layout_alignTop="@+id/button"
```

```
    android:layout_centerHorizontal="true"
    android:onClick="zoom"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="fade"
    android:id="@+id/button3"
    android:layout_alignTop="@+id/button2"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:onClick="fade"/>

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="blink"
    android:onClick="blink"
    android:id="@+id/button4"
    android:layout_below="@+id/button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="move"
    android:onClick="move"
    android:id="@+id/button5"
    android:layout_below="@+id/button2"
    android:layout_alignRight="@+id/button2"
    android:layout_alignEnd="@+id/button2"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignStart="@+id/button2" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="slide"
    android:onClick="slide"
    android:id="@+id/button6"
    android:layout_below="@+id/button3"
    android:layout_toRightOf="@+id/textView"
    android:layout_toEndOf="@+id/textView" />

</RelativeLayout>
```

res/anim/myanimation.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="5000"
    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="5000"
    android:pivotX="50%"
    android:pivotY="50%" >
  </scale>

</set>
```

res/anim/clockwise.xml

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

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

  <rotate xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:startOffset="5000"
    android:fromDegrees="360"
    android:toDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="5000" >
  </rotate>

</set>
```

res/anim/fade.xml

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

  <alpha
    android:fromAlpha="0"
    android:toAlpha="1"
    android:duration="2000" >
  </alpha>

  <alpha
    android:startOffset="2000"
    android:fromAlpha="1"
    android:toAlpha="0"
    android:duration="2000" >
  </alpha>

</set>
```

res/anim/blink.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <alpha android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:duration="600"
    android:repeatMode="reverse"
    android:repeatCount="infinite"/>
</set>
```

res/anim/move.xml

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

  <translate
    android:fromXDelta="0%p"
    android:toXDelta="75%p"
    android:duration="800" />
</set>
```

res/anim/slide.xml

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

  <scale
    android:duration="500"
    android:fromXScale="1.0"
    android:fromYScale="1.0"
    android:interpolator="@android:anim/linear_interpolator"
    android:toXScale="1.0"
    android:toYScale="0.0" />
</set>
```

res/values/string.xml

```
<resources>
  <string name="app_name">My Application</string>
</resources>
```

AndroidManifest.xml

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

```
  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
      android:name="com.example.animation.MainActivity"
      android:label="@string/app_name" >

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

    </activity>

  </application>
</manifest>
```

Session 06

Goal

To know about Google Maps for Android applications

Google Maps for Android applications

Android allows to integrate google maps in application. Maps can be customized.

Google Map - Layout file

Map fragment has to be added into xml layout file. Its syntax is given below −

```
<fragment
  android:id="@+id/map"
  android:name="com.google.android.gms.maps.MapFragment"
  android:layout_width="match_parent"
  android:layout_height="match_parent"/>
```

Google Map - AndroidManifest file

The next thing is to add some permissions along with the Google Map API key in the AndroidManifest.XML file. Its syntax is given below −

```
<!--Permissions-->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="com.google.android.providers.gsf.permission.
  READ_GSERVICES" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<!--Google MAP API key-->
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
```

```
  android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0" />
```

Customizing Google Map

Google map can easily be customized from its default view & changed according to demand.

Adding Marker

```
final LatLng TutorialsPoint = new LatLng(21 , 57);
Marker TP = googleMap.addMarker(new MarkerOptions()
  .position(TutorialsPoint).title("TutorialsPoint"));
```

Changing Map Type

```
googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
```

Enable/Disable zoom

```
googleMap.getUiSettings().setZoomGesturesEnabled(true);
```