

Practical No: 16

Exercise

```
package com.example.datetimepicker;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TimePicker;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {

    Button changeTime, changeDate;
    EditText timeET, dateEt;

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

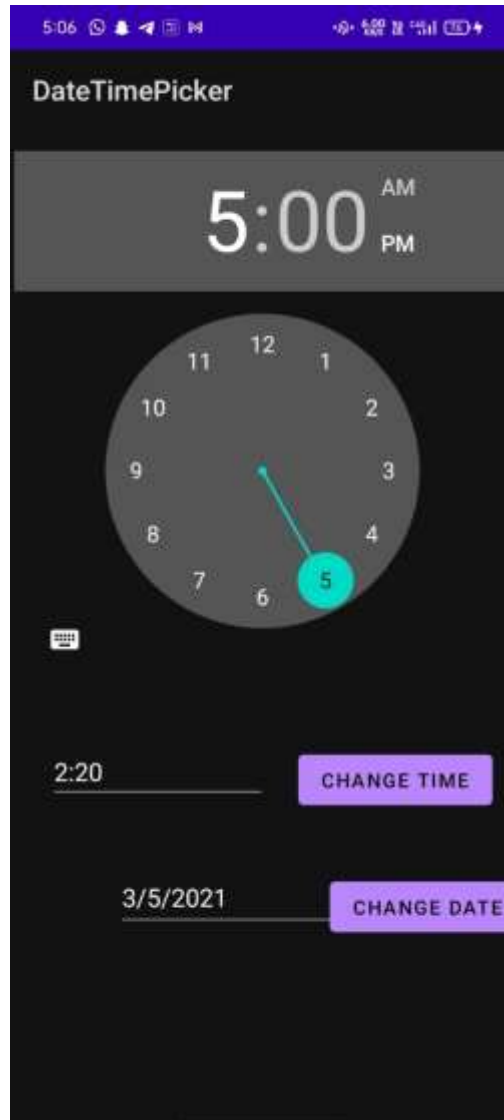
        changeTime = (Button) findViewById(R.id.changetimebtn);
        changeDate = (Button) findViewById(R.id.button2);
        changeTime.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                timeET = (EditText) findViewById(R.id.editTextTime);
                Calendar mcurrentTime = Calendar.getInstance();
                int hour = mcurrentTime.get(Calendar.HOUR_OF_DAY);
                int minute = mcurrentTime.get(Calendar.MINUTE);
                TimePickerDialog mTimePicker;
                mTimePicker = new TimePickerDialog(MainActivity.this, new
                TimePickerDialog.OnTimeSetListener() {
                    @Override
                    public void onTimeSet(TimePicker timePicker, int selectedHour,
                    int selectedMinute) {
                        timeET.setText( selectedHour + ":" + selectedMinute);
                    }
                }, hour, minute, true); //Yes 24 hour time
                mTimePicker.setTitle("Select Time");
                mTimePicker.show();
            }
        });

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

Practical No: 16

```
dateEt = (EditText) findViewById(R.id.editTextDate);
Calendar c=Calendar.getInstance();
Integer month=c.get(Calendar.MONTH);
Integer day=c.get(Calendar.DAY_OF_MONTH);
Integer year=c.get(Calendar.YEAR);

DatePickerDialog datePickerDialog =new
DatePickerDialog(MainActivity.this, new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int month, int
dayOfMonth) {
        dateEt.setText(dayOfMonth+"/"+month+"/"+year);
    }
},year,month,day);
datePickerDialog.show();
});
}
```



Practical No: 16

Practical Related Questions

1. XML time picker tag with its attributes

```
<TimePicker
    android:id="@+id/timePicker1"
    android:layout_width="354dp"
    android:layout_height="356dp"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.491"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

2. Write and explain methods of timepicker class

1	is24HourView() This method returns true if this is in 24 hour view else false
2	isEnabled() This method returns the enabled status for this view
3	setCurrentHour(Integer currentHour) This method sets the current hour
4	setCurrentMinute(Integer currentMinute) This method sets the current minute
5	setEnabled(boolean enabled) This method set the enabled state of this view
6	setIs24HourView(Boolean is24HourView) This method set whether in 24 hour or AM/PM mode
7	setOnTimeChangeListener(TimePicker.OnTimeChangeListener onTimeChangeListener) This method Set the callback that indicates the time has been adjusted by the user

Practical No: 16

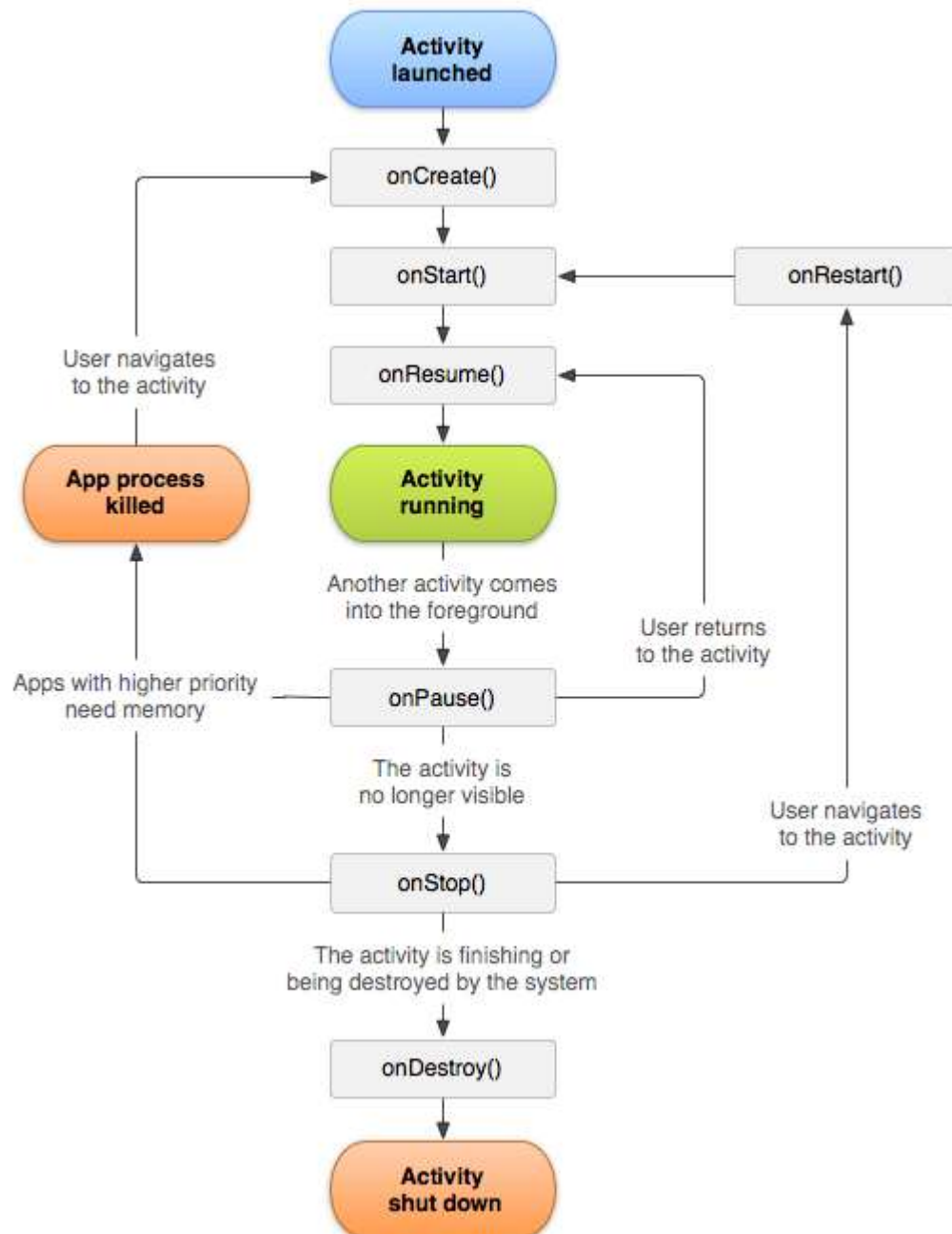
3. List and explain any 5 methods of Date Picker class

Sr.No	Method & description
1	getDayOfMonth() This method gets the selected day of month
2	getMonth() This method gets the selected month
3	getYear() This method gets the selected year
4	setMaxDate(long maxDate) This method sets the maximal date supported by this DatePicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone
5	setMinDate(long minDate) This method sets the minimal date supported by this NumberPicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone

Practical 17

Practical Related Questions

- 1. Draw the activity life cycle diagram.**



Practical 17

2. Give the hierarchy of directory structure where you store activity file.

res/layout/activity_foo.xml - This file describes the layout of the activity's UI. This means the placement of every view object on one app screen.

src/.../FooActivity.java - The Activity "controller" that constructs the activity using the view, and handles all event handling and view logic for one app screen.

3. Write difference between onStop() and onDestroy() methods, also between onPause() and onResume() methods.

onDestroy() is called whenever:

- The user takes out the activity from the "recent apps" screen.
- The user takes out the activity from the "recent apps" screen.

onStop() is called whenever:

- The user leaves the current activity.

So in your example, when the user launches Activity B, Activity A called onStop().

Exercise

1.

a) MainActivity.java

```
package com.example.activitylife;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }

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

Practical 17

```
    Log.d("lifecycle", "onStart invoked");
}

@Override
protected void onResume(){
    super.onResume();
    Log.d("lifecycle", "OnResume Invoked");
}

@Override
protected void onPause(){
    super.onPause();
    Log.d("lifecycle", "OnPause Executed");
}

@Override
protected void onStop(){
    super.onStop();
    Log.d("lifecycle", "OnStop Lifecycle Executed");
}

@Override
protected void onRestart(){
    super.onRestart();
    Log.d("lifecycle", "OnRestart Executed");
}

@Override
protected void onDestroy(){
    super.onDestroy();
    Log.d("lifecycle", "OnDestroy Executed");
}
}
```

b) activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

Practical no: 19

Practical Related Questions

1. Write in detail which methods are needed to implement Content Provider class.

Following are the six abstract methods and their description which are essential to override as the part of ContentProvider class:

query()	A method that accepts arguments and fetches the data from the desired table. Data is returned as a cursor object.
insert()	To insert a new row in the database of the content provider. It returns the content URI of the inserted row.
update()	This method is used to update the fields of an existing row. It returns the number of rows updated.
delete()	This method is used to delete the existing rows. It returns the number of rows deleted.
getType()	This method returns the Multipurpose Internet Mail Extension (MIME) type of data to the given Content URI.
onCreate()	As the content provider is created, the android system calls this method immediately to initialise the provider.

Practical no: 19

2. Explain different parts of an URI in android application. Also write the format of URI.

URI syntax and components

At the highest level a URI reference (hereinafter simply "URI") in string form has the syntax

```
[scheme : ]scheme-specific-part[ # fragment]
```

where square brackets [...] delineate optional components and the characters `:` and `#` stand for themselves.

An *absolute* URI specifies a scheme; a URI that is not absolute is said to be *relative*. URIs are also classified according to whether they are *opaque* or *hierarchical*.

An *opaque* URI is an absolute URI whose scheme-specific part does not begin with a slash character (`'/'`). Opaque URIs are not subject to further parsing. Some examples of opaque URIs are:

```
mailto:java-net@java.sun.com
```

```
news:comp.lang.java
```

```
urn:isbn:096139210x
```

3. Write steps to create a content provider in android applications.

- Create a class that extends `ContentProvider`.
- Create a contract class.
- Create the `UriMatcher` definition.
- Implement the `onCreate()` method.
- Implement the `getType()` method.
- Implement the CRUD methods.
- Add the content provider to your `AndroidManifest.xml`.

Practical no: 19

Exercise MainActivity.java

```
package com.example.contentprovidereg;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.CursorAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

    @Override
    public boolean onTouchEvent(MotionEvent event){
        InputMethodManager imm =
        (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);
        return true;
    }

    public void onClickAddDetails(View view){
        ContentValues values = new ContentValues();

        values.put(MyContentProvider.name, ((EditText)
        findViewById(R.id.textName)).getText().toString());

        getResolver().insert(MyContentProvider.CONTENT_URI, values);

        Toast.makeText(getBaseContext(), "New Record Inserted",
        Toast.LENGTH_LONG).show();
    }

    public void onClickShowDetails(View view){
        TextView result = (TextView) findViewById(R.id.res);
    }
}
```

Practical no: 19

```
        Cursor c =
getContentResolver().query(Uri.parse("content://com.demo.user.provider/users"), null,
null, null, null);

        if(c.moveToFirst()){
            StringBuilder str = new StringBuilder();
            while(!c.isAfterLast()){
                str.append("\n" + c.getString(c.getColumnIndex("id")) + ":- " +
c.getString(c.getColumnIndex("name")));
                c.moveToNext();
            }
            result.setText(str);
        }
        else{
            result.setText("No Records Found");
        }
    }
}
```

MyContentProvider.java

```
package com.example.contentprovidereg;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.view.inputmethod.InputMethodManager;
import android.widget.CursorAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

    @Override
    public boolean onTouchEvent(MotionEvent event){
        InputMethodManager imm =
(InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
```

Practical no: 19

```
        imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);
        return true;
    }

    public void onClickAddDetails(View view){
        ContentValues values = new ContentValues();

        values.put(MyContentProvider.name, ((EditText)
findViewById(R.id.textName)).getText().toString());

        getContentResolver().insert(MyContentProvider.CONTENT_URI, values);

        Toast.makeText(getBaseContext(), "New Record Inserted",
Toast.LENGTH_LONG).show();
    }

    public void onClickShowDetails(View view){
        TextView result = (TextView) findViewById(R.id.res);

        Cursor c =
getContentResolver().query(Uri.parse("content://com.demo.user.provider/users"), null,
null, null, null);

        if(c.moveToFirst()){
            StringBuilder str = new StringBuilder();
            while(!c.isAfterLast()){
                str.append("\n" + c.getString(c.getColumnIndex("id")) + ":- " +
c.getString(c.getColumnIndex("name")));
                c.moveToNext();
            }
            result.setText(str);
        }
        else{
            result.setText("No Records Found");
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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:background="#168BC34A"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
```

Practical no: 19

```
android:layout_height="wrap_content"
android:layout_centerVertical="true"
android:orientation="vertical"
app:layout_constraintBottom_toTopOf="@+id/imageView"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.13"
tools:ignore="MissingConstraints">

<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp"
    android:layout_marginBottom="70dp"
    android:text="@string/heading"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textColor="@android:color/holo_green_dark"
    android:textSize="36sp"
    android:textStyle="bold" />

<EditText
    android:id="@+id/textName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="40dp"
    android:hint="@string/hintText" />

<Button
    android:id="@+id/insertButton"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="20dp"
    android:background="#4CAF50"
    android:onClick="onClickAddDetails"
    android:text="@string/insertButtontext"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1"
    android:textColor="#FFFFFF"
    android:textStyle="bold" />

<Button
    android:id="@+id/loadButton"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginStart="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="20dp"
```

Practical no: 19

```
        android:layout_marginBottom="20dp"
        android:background="#4CAF50"
        android:onClick="onClickShowDetails"
        android:text="@string/loadButtonText"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Display1"
        android:textColor="#FFFFFF"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/res"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:clickable="false"
        android:ems="10"
        android:textColor="@android:color/holo_green_dark"
        android:textSize="18sp"
        android:textStyle="bold" />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/liv">
    </ListView>

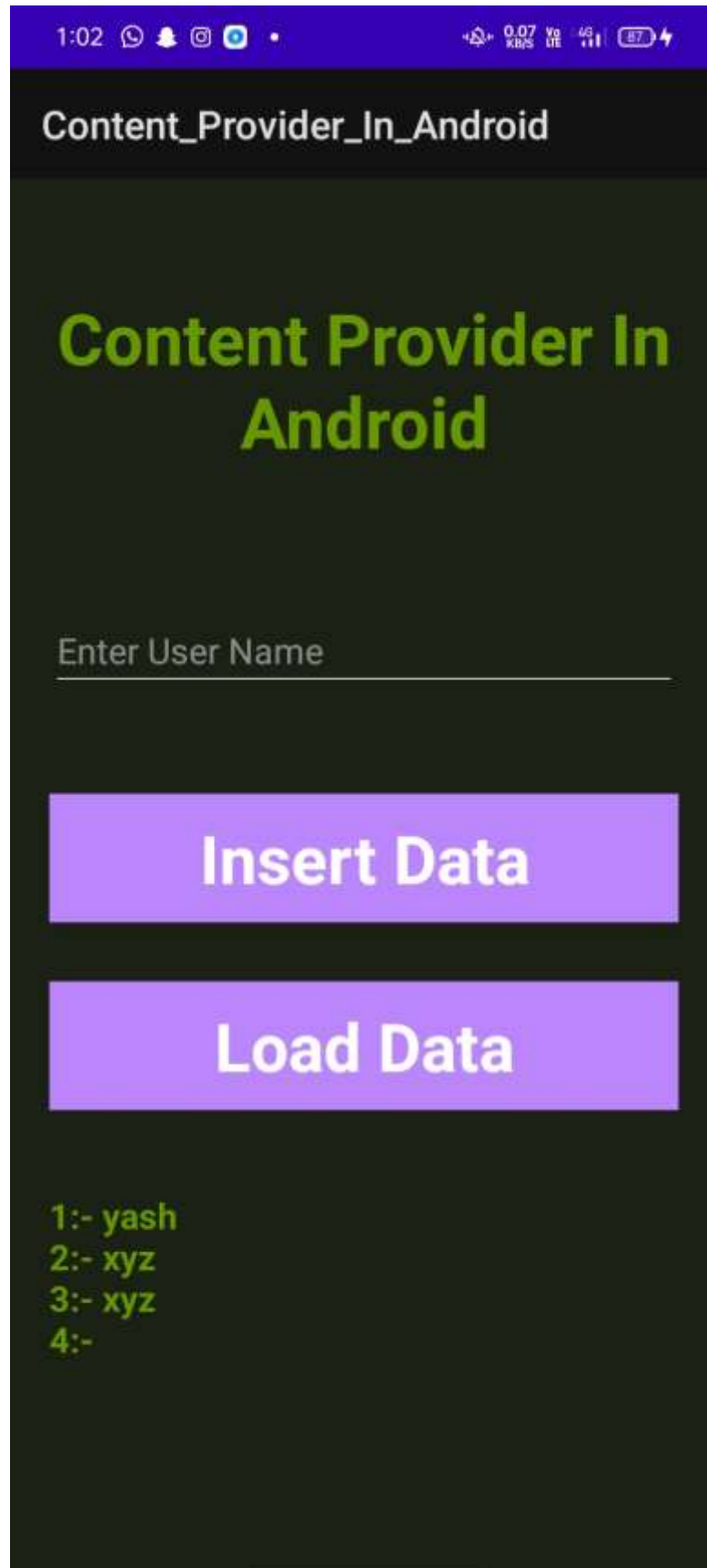
</LinearLayout>

<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    />

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

Practical no: 19

Output



Practical no: 21

Practical Related Questions

1. Differentiated between Activity Intent and Broadcasting Intent.

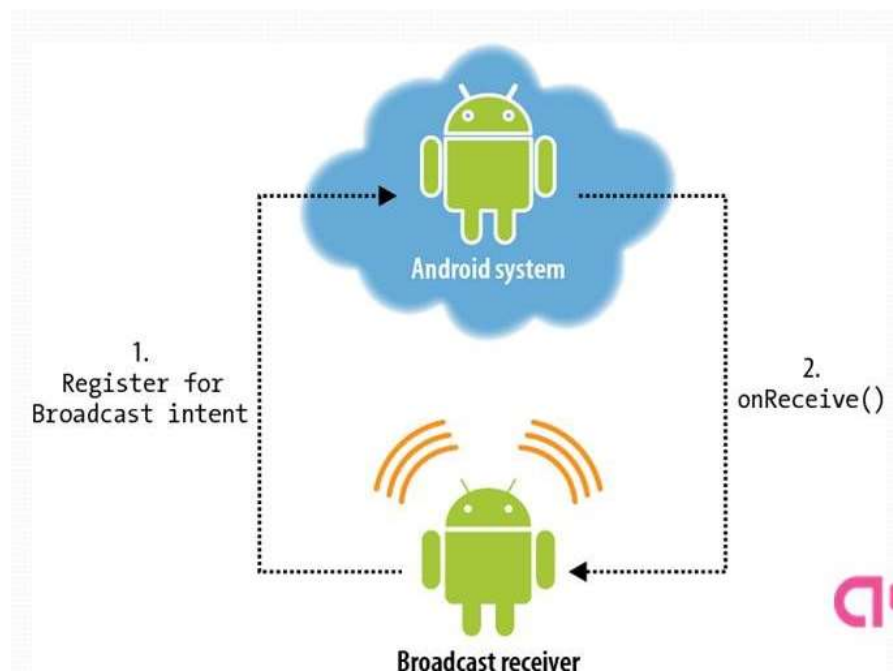
Broadcasting Intents: If you want your application itself should generate and send custom intents then you will have to create and send those intents by using the `sendBroadcast()` method inside your activity class. If you use the `sendStickyBroadcast(Intent)` method, the Intent is sticky, meaning the Intent you are sending stays around after the broadcast is complete.

Activity 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. So, it can be described as the intention to do action.

2. Draw Broadcast Receivers Lifecycle.



Practical no: 21

3. List the System Events related to Broadcast Receivers

1	android.intent.action.BATTERY_CHANGED Sticky broadcast containing the charging state, level, and other information about the battery.
2	android.intent.action.BATTERY_LOW Indicates low battery condition on the device.
3	android.intent.action.BATTERY_OKAY Indicates the battery is now okay after being low.
4	android.intent.action.BOOT_COMPLETED This is broadcast once, after the system has finished booting.
5	android.intent.action.BUG_REPORT Show activity for reporting a bug.
6	android.intent.action.CALL Perform a call to someone specified by the data.
7	android.intent.action.CALL_BUTTON The user pressed the "call" button to go to the dialer or other appropriate UI for placing a call.
8	android.intent.action.DATE_CHANGED The date has changed.
9	android.intent.action.REBOOT Have the device reboot.

Practical no: 21

Exercise MainActivity.java

```
package com.example.broadcastreceg;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button b = (Button) findViewById(R.id.button);
        b.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                Intent i = new Intent();
                i.setAction("com.broadcastreceg.CUSTOM_INTENT");
                sendBroadcast(i);
            }
        });
    }
}
```

MyReceiver.java

```
package com.example.broadcastreceg;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;
public class MyReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, " Received", Toast.LENGTH_LONG).show();
    }
}
```

AndroidManifest.xml

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

Practical no: 21

```
<receiver
    android:name=".MyReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="com.broadcastreceg.CUSTOM_INTENT">
        </action>
        <action
android:name="android.intent.action.ACTION_POWER_CONNECTED"></action>
        <action
android:name="android.intent.action.ACTION_POWER_DISCONNECTED"></action>
        <action
android:name="android.intent.action.BATTERY_CHANGED"></action>
    </intent-filter>
</receiver>
<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"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Broadcast Receiver"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Broadcast"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Practical no: 23

Practical Related Questions

1. List all the methods related to camera class.

Sr.No	Activity function description
1	startActivityForResult(Intent intent, int requestCode, Bundle options) It starts an activity , but can take extra bundle of options with it
2	startActivityFromChild(Activity child, Intent intent, int requestCode) It launch the activity when your activity is child of any other activity
3	startActivityFromChild(Activity child, Intent intent, int requestCode, Bundle options) It work same as above , but it can take extra values in the shape of bundle with it
4	startActivityFromFragment(Fragment fragment, Intent intent, int requestCode) It launches activity from the fragment you are currently inside
5	startActivityFromFragment(Fragment fragment, Intent intent, int requestCode, Bundle options) It not only launches the activity from the fragment , but can take extra values with it

Practical no: 23

Exercise

MainActivity.java

```
package com.example.cameracentricapp;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;

import java.io.File;

public class MainActivity extends AppCompatActivity {

    private ImageView imgvw;
    private Button captureBtn, videoBtn;
    private VideoView videovw;
    private static final int Image_Capture_Code = 1;
    private static final int VIDEO_CAPTURE = 101;
    Uri videoUri;

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

        imgvw = (ImageView) findViewById(R.id.imageView);
        captureBtn = (Button) findViewById(R.id.button);
        videoBtn = (Button) findViewById(R.id.button2);
        videovw = (VideoView) findViewById(R.id.videoView);

        MediaController mediaController= new MediaController(this);
        mediaController.setAnchorView(videovw);
        videovw.setMediaController(mediaController);

        captureBtn.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                Intent cam = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cam, Image_Capture_Code);
            }
        })
    }
}
```

Practical no: 23

```
});
videoBtn.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view){
        File mediaFile =
            new File(Environment.getExternalStorageDirectory()
                + "/myvideo.mp4");
        if (Build.VERSION.SDK_INT < 24) {
            videoUri = Uri.fromFile(mediaFile);
        } else {
            videoUri = Uri.parse(mediaFile.getPath()); // My work-around for
new SDKs, worked for me in Android 10 using Solid Explorer Text Editor as the
external editor.
        }

        Intent cam = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);

        cam.putExtra(MediaStore.EXTRA_OUTPUT, videoUri);
        startActivityForResult(cam, VIDEO_CAPTURE);
    }
});
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){
    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode == Image_Capture_Code){
        if(resultCode == RESULT_OK){
            videovw.setVisibility(View.INVISIBLE);
            Bitmap bp = (Bitmap) data.getExtras().get("data");
            imgvw.setVisibility(View.VISIBLE);
            imgvw.setImageBitmap(bp);
        }
        else if(resultCode == RESULT_CANCELED){
            imgvw.setVisibility(View.INVISIBLE);
            videovw.setVisibility(View.INVISIBLE);
            Toast.makeText(this, "Cancelled", Toast.LENGTH_LONG).show();
        }
    }
    else if(requestCode == VIDEO_CAPTURE){
        if(resultCode == RESULT_OK){
            videovw.setVisibility(View.VISIBLE);
            imgvw.setVisibility(View.INVISIBLE);
            videovw.setVideoURI(videoUri);

            videovw.start();
        }
        else if(resultCode == RESULT_CANCELED){
            imgvw.setVisibility(View.INVISIBLE);
            videovw.setVisibility(View.INVISIBLE);
            Toast.makeText(this, "Cancelled", Toast.LENGTH_LONG).show();
        }
    }
}
}
```

Practical no: 23

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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">

    <Button
        android:id="@+id/button"
        android:layout_width="171dp"
        android:layout_height="65dp"
        android:text="Take Picture"
        app:layout_constraintBottom_toTopOf="@+id/button2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="275dp"
        android:layout_height="280dp"
        android:visibility="invisible"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:layout_editor_absoluteY="140dp"
        tools:srcCompat="@tools:sample/avatars" />

    <Button
        android:id="@+id/button2"
        android:layout_width="169dp"
        android:layout_height="55dp"
        android:layout_marginBottom="32dp"
        android:text="Capture Video"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.504"
        app:layout_constraintStart_toStartOf="parent" />

    <VideoView
        android:id="@+id/videoView"
        android:layout_width="295dp"
        android:layout_height="298dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.653" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Practical No: 25

Practical Related Questions

1. Write the steps to perform Tween Animation.

- a) Step 1: Open or Start an Android Project.
- b) Step 2: Create the Sky Shape Drawable.
- c) Step 3: Create the Sun Shape Drawable.
- d) Step 4: Create the Grass Shape Drawable.
- e) Step 5: Include the Drawables in the Layout.
- f) Step 6: Define the Sun Rise Animation.
- g) Step 7: Apply the Animation.

2. Explain the use of from XScale and from YScale method in detail.

This method is used to define initial position, from where to start scale animation.

Exercise

MainActivity.java

```
package com.example.animationeg;

import androidx.appcompat.app.AppCompatActivity;

import android.icu.number.Scale;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.view.animation.RotateAnimation;
import android.view.animation.ScaleAnimation;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    ImageView img;
    ScaleAnimation scale;
    RotateAnimation rotate;
    AlphaAnimation alpha;
    Button b1, b2, b3, b4, b5, b6;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img = (ImageView) findViewById(R.id.imageView);

        b1 = (Button) findViewById(R.id.anticlockwise);
        b2 = (Button) findViewById(R.id.rclockwise);
        b3 = (Button) findViewById(R.id.fadein);
```


Practical No: 25

```
b4 = (Button) findViewById(R.id.fadeout);
b5 = (Button) findViewById(R.id.zoomin);
b6 = (Button) findViewById(R.id.zoomout);

b1.setOnClickListener(this);
b2.setOnClickListener(this);
b3.setOnClickListener(this);
b4.setOnClickListener(this);
b5.setOnClickListener(this);
b6.setOnClickListener(this);

}

@Override
public void onClick(View v){
    switch(v.getId()){
        case R.id.anticlockwise:
            rotate = new RotateAnimation(360, 0);
            rotate.setDuration(1000);
            img.startAnimation(rotate);
            break;
        case R.id.rclockwise:
            rotate = new RotateAnimation(0, 360);
            rotate.setDuration(1000);
            img.startAnimation(rotate);
            break;
        case R.id.fadein:
            alpha = new AlphaAnimation(0,1000);
            alpha.setDuration(1000);
            img.startAnimation(alpha);
            break;
        case R.id.fadeout:
            alpha = new AlphaAnimation(1000,0);
            alpha.setDuration(1000);
            img.startAnimation(alpha);
            break;
        case R.id.zoomin:
            scale = new ScaleAnimation(0, 2, 0, 2);
            scale.setDuration(1000);
            img.startAnimation(scale);
            break;
        case R.id.zoomout:
            scale = new ScaleAnimation(2, 0, 2, 0);
            scale.setDuration(1000);
            img.startAnimation(scale);
            break;
    }
}
```

Practical No: 25

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="117dp"
        android:layout_height="121dp"
        android:layout_marginTop="128dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:drawable/btn_star_big_on" />

    <Button
        android:id="@+id/rclockwise"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="rotate clockwise"
        app:layout_constraintBottom_toTopOf="@+id/fadein"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/anticlockwise"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="anti clockwise"
        app:layout_constraintBottom_toTopOf="@+id/fadeout"
        app:layout_constraintEnd_toEndOf="parent" />

    <Button
        android:id="@+id/fadein"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fade In"
        app:layout_constraintBottom_toTopOf="@+id/zoomin"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/fadeout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fade Out"
        app:layout_constraintBottom_toTopOf="@+id/zoomout"
        app:layout_constraintEnd_toEndOf="parent" />

    <Button
```

Practical No: 25

```
        android:id="@+id/zoomin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Zoom IN"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/zoomout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Zoom Out"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Practical No: 27

Practical Related Questions

1. Explain the use of equals() function.

The **equals()** method compares two strings, and returns true if the strings are **equal**, and false if not.

2. List the important functions which are related to GUI component “Button”.

Attribute	Description
android:id	It is used to uniquely identify the control
android:gravity	It is used to specify how to align the text like left, right, center, top, etc.
android:text	It is used to set the text.
android:textColor	It is used to change the color of text.
android:textSize	It is used to specify the size of the text.
android:textStyle	It is used to change the style (bold, italic, bolditalic) of text.
android:background	It is used to set the background color for button control.
android:padding	It is used to set the padding from left, right, top and bottom.
android:drawableBottom	It's drawable to be drawn to the below of text.
android:drawableRight	It's drawable to be drawn to the right of text.
android:drawableLeft	It's drawable to be drawn to the left of the text.

3. State the uses of Toast message.

Toasts display brief, temporary notifications. They are meant to be noticed without disrupting a user's experience or requiring an action to be taken.

Practical No: 27

Exercise

MainActivity.java

```
package com.example.logineg;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    EditText unnm, pw;
    Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = (Button) findViewById(R.id.button);
        unnm = (EditText) findViewById(R.id.editTextTextPersonName);
        pw = (EditText) findViewById(R.id.editTextTextPassword);

        b.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                String user = (String) unnm.getText().toString();
                String pass = (String) pw.getText().toString();
                Toast.makeText(MainActivity.this, user + " " + pass,
                    Toast.LENGTH_LONG).show();

                if(user.equals("abcd") && pass.equals("abc123")){
                    Toast.makeText(MainActivity.this, "Login Successful",
                        Toast.LENGTH_LONG).show();
                }
                else{
                    Toast.makeText(MainActivity.this, "Username and password does not
match", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}
```

activity_main.xml

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

Practical No: 27

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<Button
    android:id="@+id/button"
    android:layout_width="273dp"
    android:layout_height="56dp"
    android:layout_marginBottom="204dp"
    android:text="Log In"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="266dp"
    android:layout_height="40dp"
    android:ems="10"
    android:hint="Username/Email"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toTopOf="@+id/editTextTextPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.475"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.685" />

<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="273dp"
    android:layout_height="46dp"
    android:layout_marginBottom="96dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Practical No: 29

Practical Related Questions

1. **Explain the use of SmsManagerClass.**
 - SMSManager class manages operations like sending a text message, data message, and multimedia messages (MMS). For sending a text message method sendTextMessage() is used likewise for multimedia message sendMultimediaMessage() and for data message sendDataMessage() method is used.
2. **List changes that are need to be done in AndroidManifest.XML file to send and receive messages.**

We need to add permission to send and receive SMS.

We also need to add receiver class to perform certain action when SMS is received.

Practical No: 29

Exercise

MainActivity.java

```
package com.example.smsmanager;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText ph, msg;
    Button b;

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

        b = (Button) findViewById(R.id.button);
        ph = (EditText) findViewById(R.id.editTextPhone);
        msg = (EditText) findViewById(R.id.editTextTextMultiLine);

        b.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                String phone = (String) ph.getText().toString();
                String message = (String) msg.getText().toString();

                try{
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phone, null, message, null, null);
                    Toast.makeText(MainActivity.this, "SMS sent",
Toast.LENGTH_LONG).show();
                }
                catch (Exception e){
                    Toast.makeText(MainActivity.this, "Failed to send SMS: " +
e.getMessage().toString(), Toast.LENGTH_LONG).show();
                    e.printStackTrace();
                }
            }
        });
    }
}
```


Practical No: 29

SmsBroadcastReceiver.java

```
package com.example.smsmanager;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;

public class SmsBroadcastReceiver extends BroadcastReceiver {
    public static final String SMS_BUNDLE = "pdus";
    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO: This method is called when the BroadcastReceiver is receiving
        // an Intent broadcast.
        Bundle intentExtras = intent.getExtras();
        if (intentExtras != null) {
            Object[] sms = (Object[]) intentExtras.get(SMS_BUNDLE);
            String smsMessageStr = "";
            for (int i = 0; i < sms.length; ++i) {
                SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) sms[i]);

                String smsBody = smsMessage.getMessageBody().toString();
                String address = smsMessage.getOriginatingAddress();

                smsMessageStr += "SMS From: " + address + "\n";
                smsMessageStr += smsBody + "\n";
            }
            Toast.makeText(context, smsMessageStr, Toast.LENGTH_SHORT).show();
        }
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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">

    <EditText
        android:id="@+id/editTextPhone"
        android:layout_width="270dp"
        android:layout_height="56dp"
        android:ems="10"
        android:hint="Phone No"
        android:inputType="phone"
```

Practical No: 29

```
app:layout_constraintBottom_toTopOf="@+id/editTextTextMultiLine"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.489"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.871" />

<EditText
    android:id="@+id/editTextTextMultiLine"
    android:layout_width="272dp"
    android:layout_height="39dp"
    android:layout_marginBottom="68dp"
    android:ems="10"
    android:gravity="start|top"
    android:hint="Enter Message"
    android:inputType="textMultiLine"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/button"
    android:layout_width="262dp"
    android:layout_height="55dp"
    android:layout_marginBottom="304dp"
    android:text="Send SMS"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="parent" />

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

Practical No: 31

Practical Related Questions

1. List the names of map type and write the syntax to change it.

Types of Google Maps

There are four different types of Google maps, as well as an optional to no map at all. Each of them gives different view on map. These maps are as follow:

1. **Normal:** This type of map displays typical road map, natural features like river and some features build by humans.
2. **Hybrid:** This type of map displays satellite photograph data with typical road maps. It also displays road and feature labels.
3. **Satellite:** Satellite type displays satellite photograph data, but doesn't display road and feature labels.
4. **Terrain:** This type displays photographic data. This includes colors, contour lines and labels and perspective shading.
5. **None:** This type displays an empty grid with no tiles loaded.

Syntax of different types of map

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

2. Name the methods used to enable and disable zoom feature.

```
// it will enable the zoomOut button  
zoomControls.setIsZoomOutEnabled(true)  
  
// it will disable the zoomOut button  
zoomControls.setIsZoomOutEnabled(false)
```

```
// it will enable the zoomIn button  
zoomControls.setIsZoomInEnabled(true)  
  
  
// it will disable the zoomIn button  
zoomControls.setIsZoomInEnabled(false)
```

Practical No: 31

Exercise

MapsActivity.java

```
package com.example.mylocation;

import androidx.fragment.app.FragmentActivity;

import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.GooglePlayServicesUtil;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
LocationListener {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be
        used.
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync((OnMapReadyCallback) this);
    }

    @Override
    public void onLocationChanged(Location location) {
        TextView locationTv = (TextView) findViewById(R.id.LatLongLocation);
        double latitude = location.getLatitude();
        double longitude = location.getLongitude();
        LatLng latLng = new LatLng(latitude, longitude);
        mMap.addMarker(new MarkerOptions().position(latLng));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
        mMap.animateCamera(CameraUpdateFactory.zoomTo(15));
        locationTv.setText("Latitude:" + latitude + ", Longitude:" + longitude);
    }

    @Override
    public void onProviderDisabled(String provider) {
```

Practical No: 31

```
// TODO Auto-generated method stub
}

@Override
public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera.
In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be
prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the
user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    mMap.setMyLocationEnabled(true);
    LocationManager locationManager = (LocationManager)
getSystemService(LOCATION_SERVICE);
    Criteria criteria = new Criteria();
    String bestProvider = locationManager.getBestProvider(criteria, true);
    Location location = locationManager.getLastKnownLocation(bestProvider);
    if (location != null) {
        onLocationChanged(location);
    }
    locationManager.requestLocationUpdates(bestProvider, 20000, 0, this);
}

private boolean isGooglePlayServicesAvailable() {
    int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(this);
    if (ConnectionResult.SUCCESS == status) {
        return true;
    } else {
        GooglePlayServicesUtil.getErrorDialog(status, this, 0).show();
        return false;
    }
}
}
```

Practical No: 31

activity_maps.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"
    tools:context=".MapsActivity">
    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:map="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MapsActivity" ></fragment>

    <TextView
        android:id="@+id/latlongLocation"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="bottom"
        android:layout_alignParentBottom="true"
        android:background="#ff058fff"
        android:paddingTop="5dp"
        android:paddingBottom="5dp"
        android:textColor="#ffffff"
        android:paddingLeft="5dp"
        android:paddingRight="5dp" />
</RelativeLayout>
```

AndroidManifest.xml

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

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->
    <permission
        android:name="com.javapapers.currentlocationinmap.permission.MAPS_RECEIVE"
        android:protectionLevel="signature" />

    <uses-permission
        android:name="com.javapapers.currentlocationinmap.permission.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission
        android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Practical No: 31

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

<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.MyLocation">

    <!--
        The API key for Google Maps-based APIs is defined as a string resource.
        (See the file "res/values/google_maps_api.xml").
        Note that the API key is linked to the encryption key used to sign the
APK.
        You need a different API key for each encryption key, including the
release key that is used to
        sign the APK for publishing.
        You can define the keys for the debug and release targets in src/debug/
and src/release/.
    -->
    <meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="@string/google_maps_key" />

    <activity
        android:name=".MapsActivity"
        android:label="@string/title_activity_maps">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

</manifest>
```