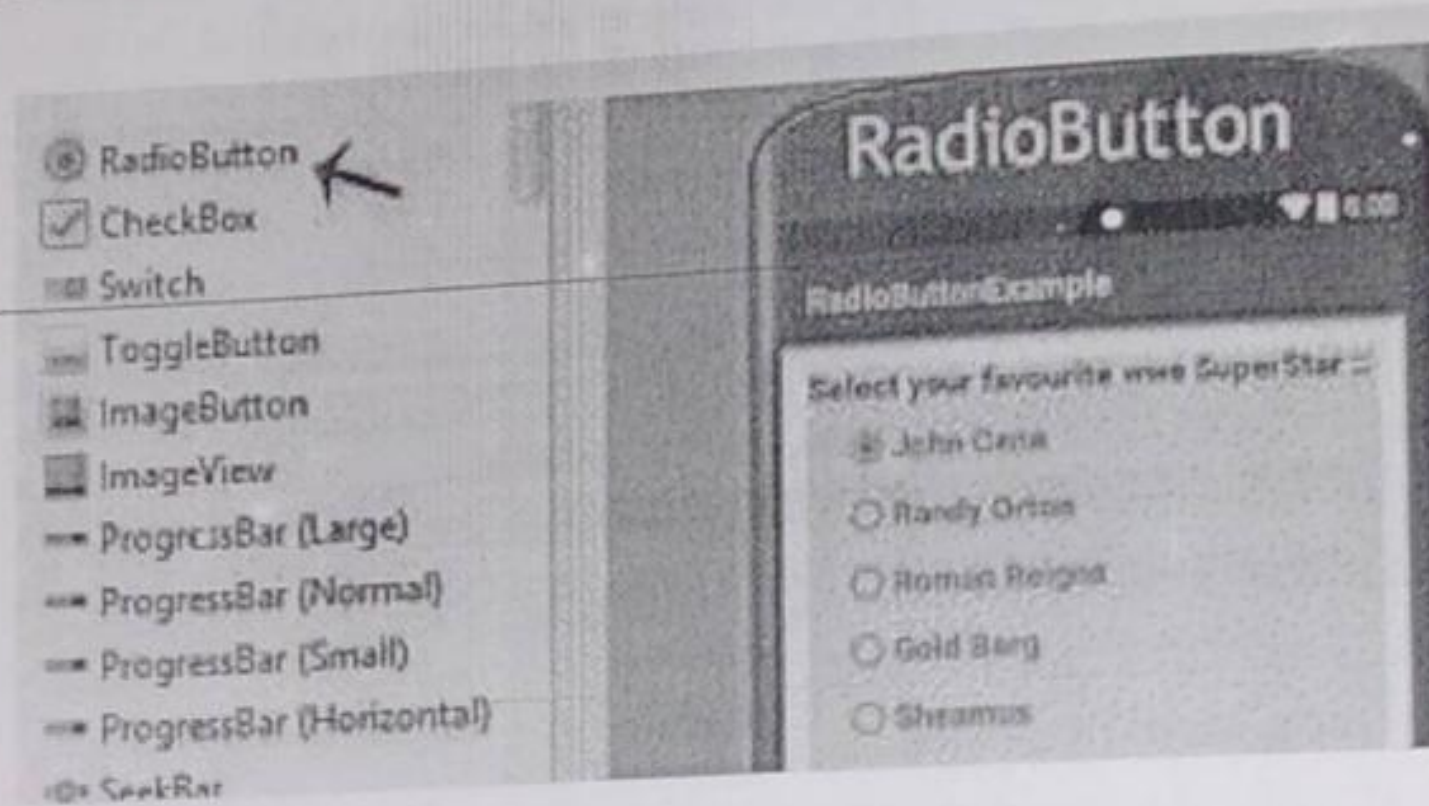


4. **setOnCheckedChangeListener()** : This registers a callback to be invoked when the checked radio button changes in this group. We must supply instance of Radio Group. OnCheckedChangeListener to setOnCheckedChangeListener() method



VIII. Resources used (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write xml to create a Radio button.
2. Write the purpose of Radio Button
3. List different methods of Radio Button

(Space for answers)

```

1) → <?xml version="1.0" encoding="utf-8"
<RadioGroup xmlns:android="http://schemas.android
.com/apk/res/android"
<RadioButton android:id="@+id/radio_pirates"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout:text="@string/pirates"
android:onClick="onRadioButtonClicked"/>

```

2) → Radio buttons allow the user to select one option from a set. You should use radio buttons for optional sets that the user needs to

are mutually exclusive.

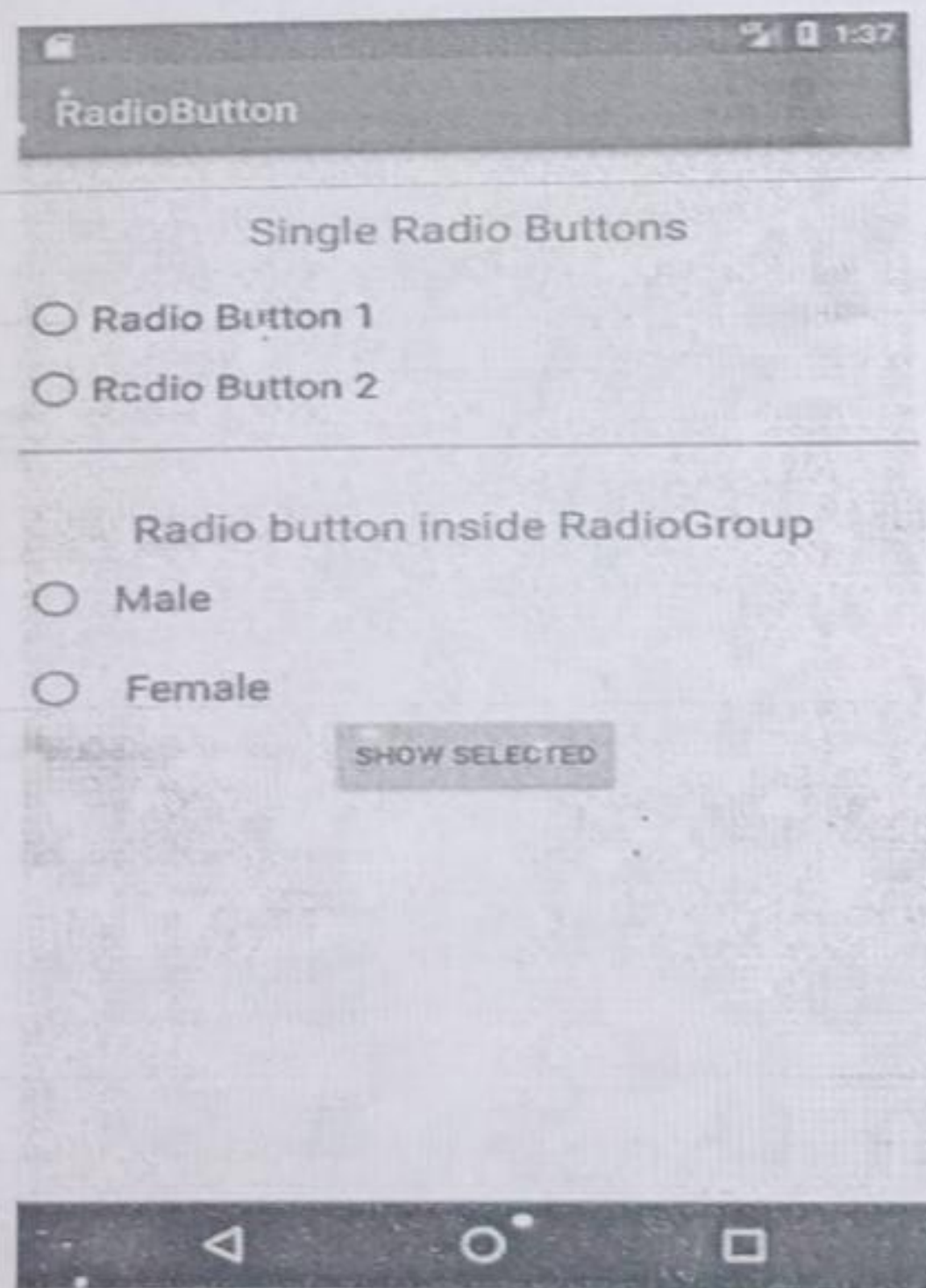
3) → checked and unchecked are methods of radiobutton

X. Exercise

Note: Faculty must ensure that every group of students use different examples.

(Use blank space for answers or attach more pages if needed)

1. Write a program to show the following output. First two radio buttons are without using radio group and next two radio buttons are using radio group. Note the changes between these two. Also toast which radio button has been selected.



(Space for answers)

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

```
xmlns:tools="{http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="example.javapoint.com.
radiobutton.MainActivity"}>
```

```
<TextView
```

```
android:id="@+id/textView1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:gravity="center_horizontal"
android:textSize="22dp"
android:text="Single Radio Buttons"/>
```

```
<!-- Default RadioButtons -->
```

```
<RadioButton
```

```
android:id="@+id/radioButton1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:text="Radio Button1"
android:layout_marginTop="20dp"
android:textSize="20dp"/>
```

```
<RadioButton
```

```
android:id="@+id/radioButton2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Radio Button2"
android:layout_marginTop="10dp"
android:textSize="20dp"/>
```

```
<View
```

```
android:layout_width="fill_parent"
android:layout_height="1dp"
```

```

android:layout_marginTop="20dp"
android:background="#BBB894" /
<TextView
android:id="@+id/textView2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
    
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

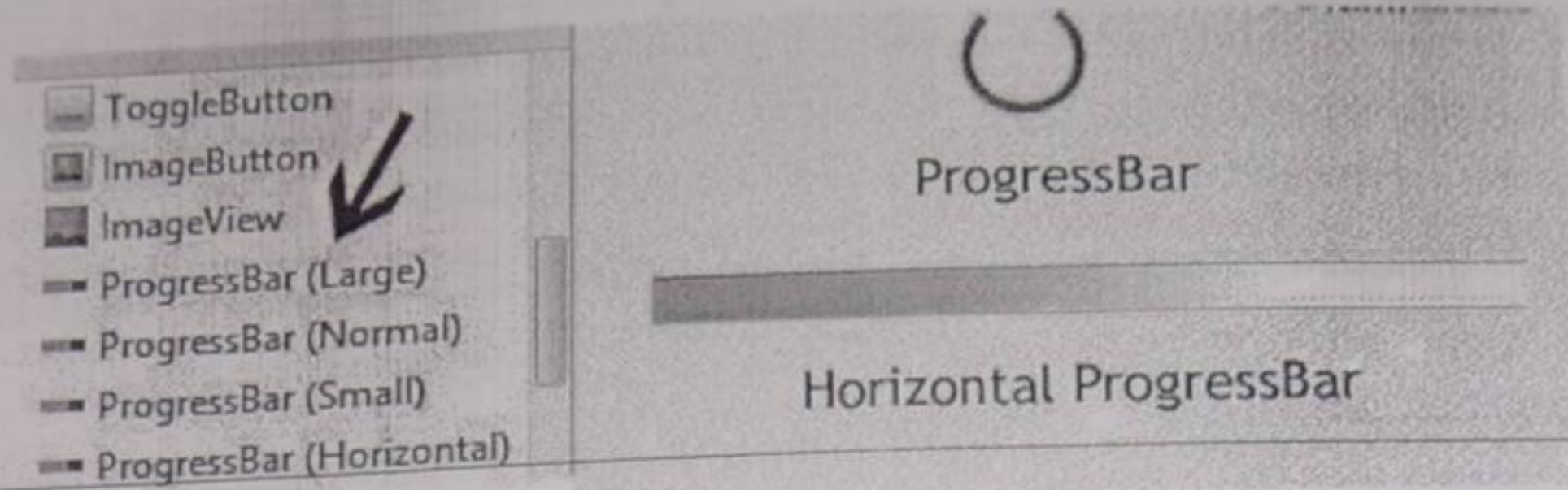
List of students/Team Members

1.
2.
3.
4.

Marks Obtained ¹			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

```
android:gravity="center-horizontal"
android:textSize="22dp"
android:text="Radio button Inside Radiogroup"/>
<Radiogroup
  android:layout_width="wrap-content"
  android:layout_height="wrap-content"
  android:id="@+id/radiogroup">
  <RadioButton
    android:id="@+id/radioMale"
    android:layout_width="fill-parent"
    android:layout_height="wrap-content"
    android:text="Male"
    android:layout_marginTop="10dp"
    android:checked="false"
    android:textSize="20dp"/>
  <RadioButton
    android:id="@+id/radiofemale"
    android:layout_width="fill-parent"
    android:layout_height="wrap-content"
    android:text="Female"
    android:layout_marginTop="20dp"
    android:Checked="false"
    android:textSize="20dp"/>
</Radiogroup>
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="ShowSelected"  
    android:id="@+id/button"  
    android:onClick="onClickbuttonMethod"  
    android:layout_gravity="center_horizontal" />  
</LinearLayout>
```



VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State different methods to update the percentage of progress displayed.
2. Write an xml tag for the determinate progress bar.
3. List different progress bar styles provided by the system.

(Space for answers)

1) → `setProgress(int)` method
 2) `incrementProgressBy(int)` method

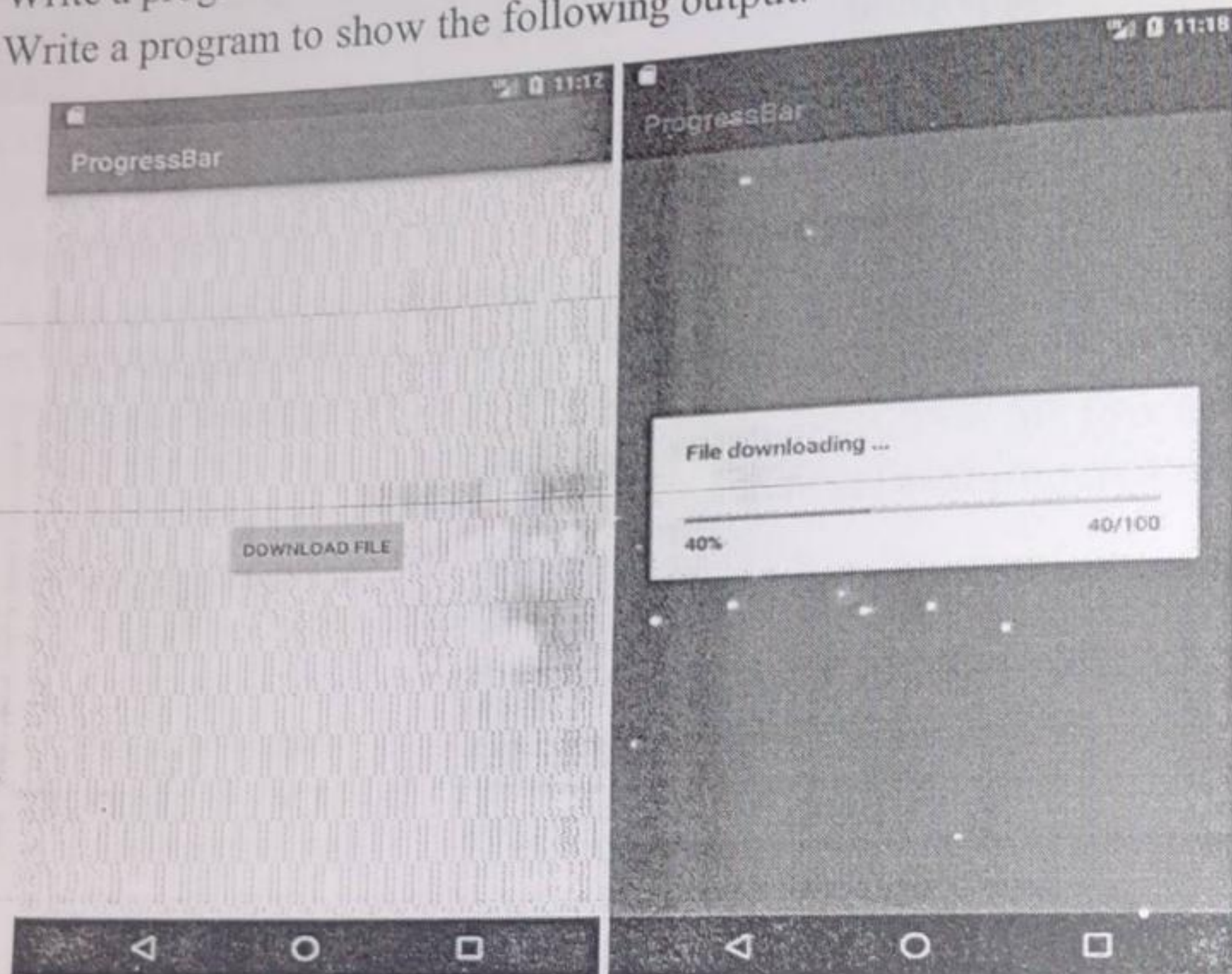
2) → `<ProgressBar`
`android:id="@id/simple_progress_bar"`
`android:layout_width="wrap_content"`
`android:layout_height="wrap_content"/>`

3) → `Widget.ProgressBar.Horizontal`
`Widget.ProgressBar.Small`
`Widget.ProgressBar.Large`
`Widget.ProgressBar.Inverse`
`Widget.ProgressBar.Small.Inverse`
`Widget.ProgressBar.Large.Inverse`

X. Exercise

Note: Below given are few sample questions for reference. Teachers must design different questions for practice.
(Use blank space for answers or attach more pages if needed)

1. Write a program to display circular progress bar.
2. Write a program to show the following output.



(Space for answers)

1) →

```

<?xml version="1.0" encoding="utf-8"?>
<include xmlns:android="http://schemas.
android.com/apk/res/android"
android:fromDegrees="270"
android:toDegrees="270">
<shape
android:innerRadiusRatio="2.5"
android:shape="ring"
android:thickness="1dp"
android:useLevel="true"
    
```



```

<gradient
  android:angle="0"
  android:endcolor="#0070D6"
  android:startcolor="#0070D6"
  android:type="sweep"
  android:uselevels="false"/>
</shape>
</rotate>

```

```

<shape xmlns:android="http://schemas-
  android.com/apk/res/android"
  android:innerRadiusRatio="2.5"
  android:shape="ring"
  android:thickness="1dp"
  android:uselevel="false"
  <solid android:color="#ccc"/>
</shape>

```

```

package app.tutorialspoint.com.sample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

```

<manifest xmlns:android="http://schemas-
  android.com/apk/res/android"
  package="app.tutorialspoint.com.sample"
  <uses-permission android:name="android-
  permission.CALL_PHONE"/>
<application

```

```

android:allowBackup="true"
android:icon="@mipmap/launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/launcher_rou
android:supportRtl="true"
android:theme="@style/AppTheme"
<activity android:name=".MainActivity">

```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

List of students/Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

```

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

```

~~MainActivity Java~~

```

1) → progressBar = new ProgressDialog (v.getContext());
progressBar.setCancelable(true);
progressBar.setMessage

```

```

2) → import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.widget.Button;
import android.view.View;
import android.view.View.OnClickListener;
public class MainActivity extends Activity {
Button startDownload;
ProgressDialog progressBar;
private int progressBarStatus=0;

```

```

private Handler progressBarHandler=new Handler();
private long level=0;
@Override
public void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
startDownload=(Button)
findViewById(R.id.startButton);
startDownload.setOnClickListener(new OnClickListener() {

```

```

@Override
public void onClick(View v){
ProgressBar=new ProgressDialog(v.getContext());
progressBar.setCancelable(true);
progressBar.setMessage("File Downloading --");
ProgressBar.setProgressStyle(ProgressDialog.STYLE_
HORIZONTAL);

```

```

ProgressBar.setProgress(0);
ProgressBar.setMax(100);
ProgressBar.show();
ProgressBarStatus=0;
level=0;

```

```

new Thread(new Runnable() {

```

```

public void run() {

```

```

while(ProgressBarStatus<100) {

```

```

ProgressBarStatus=progressLevel();

```

```

try {

```

```

Thread.sleep(1000);

```

```

}
catch

```

```

(InterruptedException) {

```

```

e.printStackTrace();

```

```

progressBarHandler.post(new Runnable() {

```

```

public void run() {

```

```

progressBar.setProgress(ProgressBarStatus)

```

```

}});

```

```

}
if(ProgressBarStatus>=100) {

```

```

try {

```

```

Thread.sleep(2000);

```

```

}
catch

```

```

(InterruptedException) {
e.printStackTrace();
runOnUiThread(new Runnable() {
public void run() {
new JLabel().setText("File Download Successfully!!!");
public int progresslevel() {
while (level <= 100) {
level++;
if (level == 10) {
return 10;
} else if (level == 20) {
return 20;
} else if (level == 30) {
return 30;
} else if (level == 40) {
return 40;
} else if (level == 50) {
return 50;
} else if (level == 60) {
return 60;
} else if (level == 70) {
return 70;
} else if (level == 80) {
return 80;
} else if (level == 90) {
return 90;
}
}
}
}
}

```

mi
P
@
pu
sc
e

```
}  
}  
return 100;  
}  
}
```

Activity_main Activity.xml

```
<LinearLayout  
    android:layout_width="match-parent"  
    android:layout_height="match-parent"  
    android:orientation="vertical"  
    android:gravity="center">  
    <Button  
        android:id="@+id/startDownload"  
        android:layout_width="250dp"  
        android:layout_height="100dp"  
        android:text="DOWNLOAD FILE" />  
    </LinearLayout>
```

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List all attributes of Image View.
2. Write steps to add following string array to grid view.
static final String [] example= new String {"A", "B", "C", "D", "E"};
3. Describe android:stretchMode attribute of Grid view in detail.

(Space for answers)

1) → 1) android: maxHeight
2) android: maxWidth
3) android: src
4) android: scaleType
5) android: tint

2) → import android.app.Activity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;
public class MainActivity extends Activity {
 GridView grid;
 static final String[] letters = new String[] {
 "A", "B", "C", "D", "E"};
 @Override
 public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.main);
 grid = (GridView) findViewById(R.id.gridView);
 ArrayAdapter adapter = new ArrayAdapter<this,
 android.R.layout.simple_list_item_1, letters>;
 grid.setAdapter(adapter);
 grid.setOnItemClickListener(new OnItemClickListener() {
 public void onItemClick(AdapterView parent, View v,
 int position, long id) {
 Toast.makeText(getApplicationContext(),
 ((TextView) v).getText().toString(), Toast.LENGTH_SHORT).show();
 }
 });
 }

} }) ; }

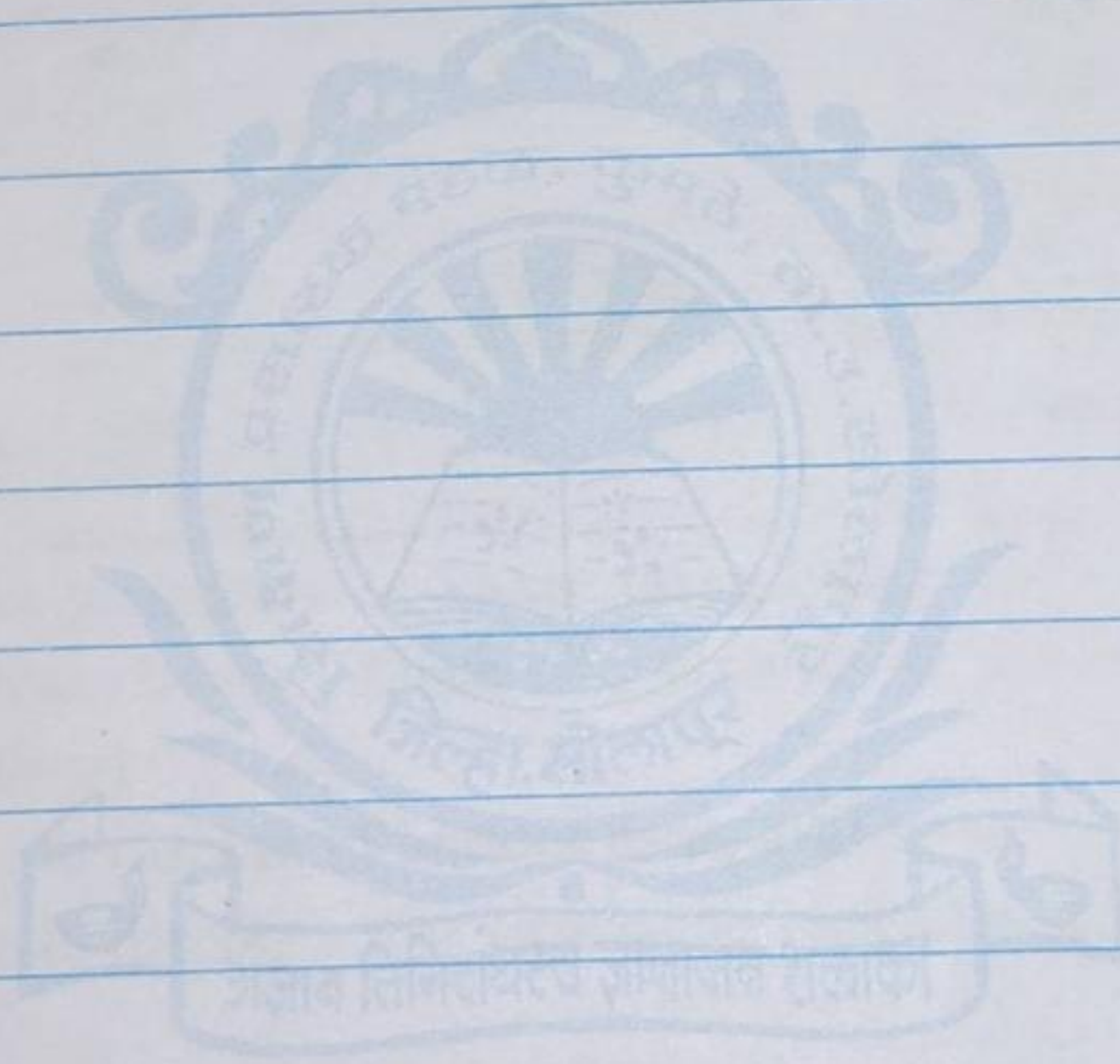
3) →

Android: Stretch Mode

Defines how columns should stretch to fill the available empty space, if any... spacing width: The spacing between each column is stretched.

ColumnWidth: Each column is stretched equally.

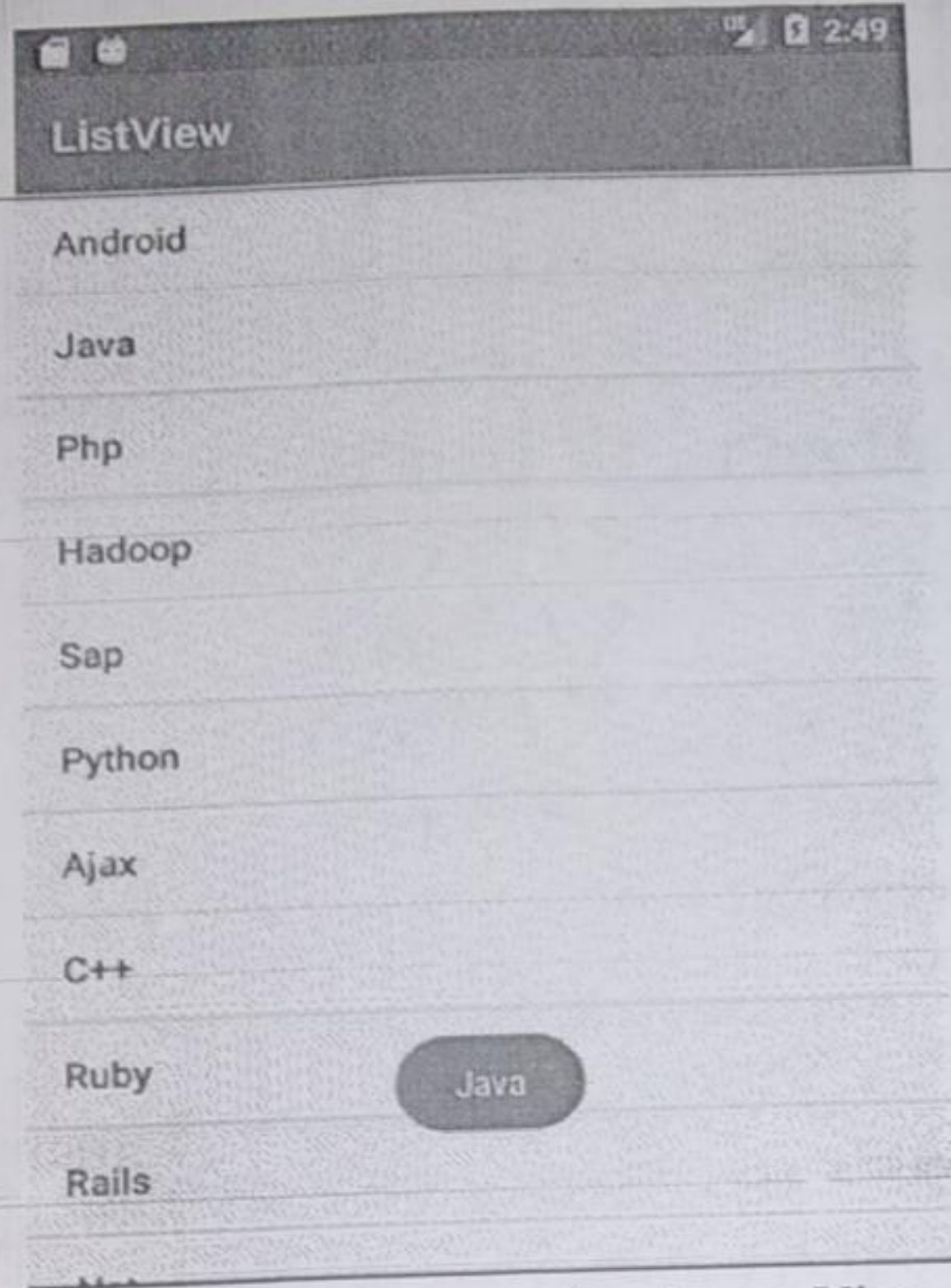
SpacingWidthUniform: The spacing between each column is uniformly stretched



X. Exercise

(Use blank space provide for answers or attached more pages if needed)

1. Write a program to show the following output. Use appropriate view for the same.



2. Write a program to display an image using Image View and a button named as "Change Image". Once you click on button another image should get displayed.
3. Write a program to display 15 buttons using grid view.
4. Write a program to display a text view using vertical scroll view.

(Space for answers)

```

1) → xml version="1.0" encoding="utf-8"
<TextView xmlns:android="https://schemas.android
.com/apk/res/android
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="MediumText"
android:textStyle="bold"
android:textAppearance="?android:attr/textAppearance
medium"
android:layout_marginLeft="10dp"
android:layout_marginTop="15dp"
android:padding="2dp"
android:textColor="#4d4d4d"
    
```

```

<resources
<String name = "app_names" ListView <String>
<String array name = "array = technology"
<item> Android </item>
<item> Java </item>
<item> PHP </item>
<item> Hadoop </item>
<item> SAP </item>
<item> python </item>
<item> ajax </item>
<item> C++ </item>
<item> Ruby </item>
<item> Rails </item>
</String array>
</resources

```

MainActivity.java

```

package listView.example.com.listView;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    ListView listView;
    TextView textView;
    String[] listItem;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView = (ListView) findViewById(R.id.listView);
        textView = (TextView) findViewById(R.id.textView);
        listItem = getResources().getStringArray(R.array - technology);
    }
}

```

```
final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, android.R.id.text1, listItem);
listView.setAdapter(adapter);
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

List of students/Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

```

public void onItemClick(AdapterView <?> adapterView, view
int position, long id) {
String value = adapter.getItem(position);
Toast.makeText(getApplicationContext(), value, Toast.LENGTH
SHORT).show();
}
}
}

```

```

-> →
package com.innosen.third // your packagename
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.app.Activity;
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}

```

```

public void mess(View v) {
ImageView im = (ImageView) findViewById(R.id.
imageView1);
// get clicked button id from view object
switch (v.getId()) {
case R.id.button1;
im.setImageResource(R.drawable.myimage2);
break;
}
}

```

```
case R.id.button2  
    im.setImageResource(R.drawable.myimage2);  
    break;  
}  
}
```

③

```
package com.example.gridview;  
import android.os.Bundle;  
import android.app.Activity;  
import android.view.Menu;  
import android.widget.AdapterView;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.AdapterView.OnItemClickListener;  
import android.widget.ArrayAdapter;  
import android.widget.GridView;  
import android.widget.TextView;  
import android.widget.Toast;  
public class MainActivity extends Activity {  
    GridView gridView;  
    static final String[] numbers = new String[] {  
        "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13",  
        "14", "15" };  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        gridView = (GridView)  
            findViewById(R.id.gridView);  
        ArrayAdapter<String> adapter = new ArrayAdapter<  
            String>
```

```

C this, android R.layout.simple_list_item_1, numbers);
gridView.setAdapter(adapter);
gridView.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?>
        parent, View view, int position, long id) {
        Toast.makeText(getApplicationContext(), ((TextView)
            view).getText(),
            Toast.LENGTH_LONG).show();
    }
});

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main,
        menu);
    return true;
}
}
}

```

4)

```

-> <?xml version="1.0" encoding="utf-8" ?>
<android.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">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="0dp"

```

```
tools:layout_editor_absoluteY="-127dp">  
<TextView  
  android:id="@id/scrolltext"  
  style="@style/AppTheme"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:text="@string/scrolltext"  
  android:textColor="@color/green"/>  
</ScrollView>  
</androidx.constraintlayout.widget.Constraint  
Layout>
```

Following is the example to create a toast.
 Toast toast = Toast.makeText(getApplicationContext(),
 "This is a message displayed in a Toast",
 Toast.LENGTH_SHORT); toast.show();

The Toast.makeText() method is a factory method which creates a Toast object. The method takes 3 parameters. First the methods needs a Context object which is obtained by calling getApplicationContext(). Note: The getApplicationContext() method is a method that exists inside activities, so the above code has to be located in an Activity subclass to work.

The second parameter is the text to be displayed in the Toast. The third parameter is the time duration the Toast is to be displayed.

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

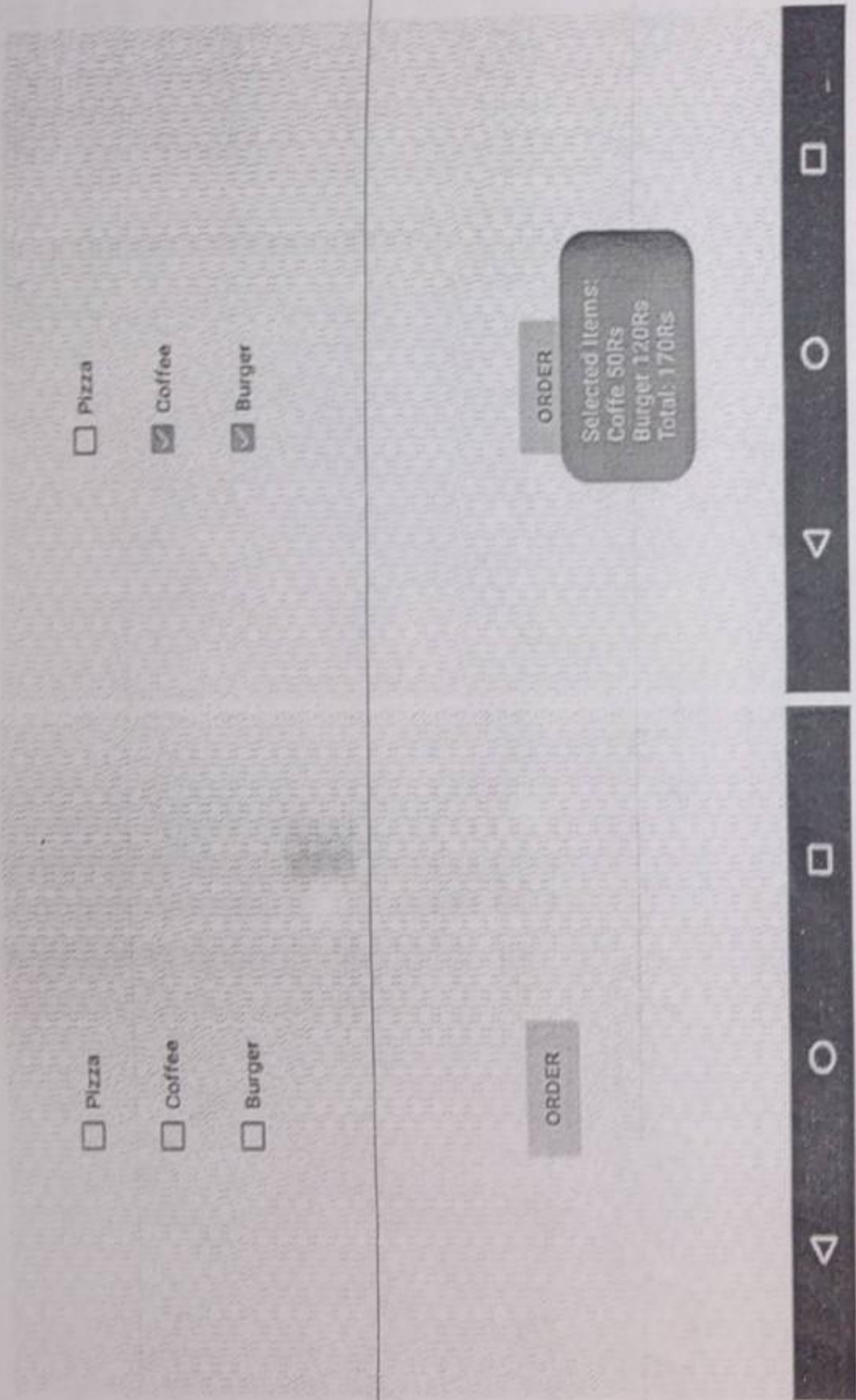
Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

- List all predefined constants to specify the overall positioning of the Toast. Which method is used to change the positioning of a Toast message on the screen?
- List two constants of Toastclass.

(Space for answers)

1) → 1) TOP
 2) BOTTOM
 3) LEFT
 4) RIGHT
 5) CENTER
 6) CENTER - HORIZONTAL
 7) CENTER - VERTICAL

2) 1) Toast.LENGTH_SHORT
 2) Toast.LENGTH_LONG



(Space for answers)

1) →
Toast toast = Toast.makeText(getApplicationContext(),
"This is a message displayed in a Toast",
Toast.LENGTH_SHORT);
toast.show();
package codedast; toast;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
 CompatActivity?;

```
Button b1;  
@Override  
protected void onCreate(Bundle savedInstanceState)  
{  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    b1 = (Button) findViewById(R.id.toast_button);  
    b1.setOnClickListener(new View.OnClickListener(  
        @Override  
        public void onClick(View v) {  
            Toast.makeText(MainActivity.this, "Message  
for you: You have got a mail",  
                Toast.LENGTH_LONG).show();  
        }  
    ));  
}
```

/* Another way to display a Toast message

```
Toast t = Toast.makeText(MainActivity.this, "Message  
for you: You have got a mail",  
    Toast.LENGTH_LONG);  
t.show();  
*/
```

2) →

activity main.xml -

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context="com.smriti.exp15-2"
    MainActivity
    tools:ignore="MergeRootFrame">
```

```
<CheckBox
```

```
    android:id="@+id/checkBox1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pizza"/>
```

```
<CheckBox
```

```
    android:id="@+id/checkBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Coffee"/>
```

```
<CheckBox
```

```
    android:id="@+id/checkBox3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Burgers"/>
```

```
</CheckBox
```

```
</CheckBox
```

```
</LinearLayout>
```

```
</activity>
```

```
</manifest>
```

```
</androidx.appcompat.app.AppCompatActivity>
```

```
</android.os.Bundle>
```

```
</android.os.Bundle>
```

Toast.makeText(MainActivity.this, "Coffee" + c + "\n Total" + t, Toast.LENGTH_SHORT).show();

if (c1.isChecked() == false && c2.isChecked() == false && p3.isChecked() == true)

Toast.makeText(MainActivity.this, "Burger" + b + "\n Total = ", Toast.LENGTH_SHORT).show();

if (c1.isChecked() == true && c2.isChecked() == true && c3.isChecked() == false)

sum = p + c;

Toast.makeText(MainActivity.this, "Pizza" + p + "\n Coffee" + c + "\n Total = " + sum, Toast.LENGTH_SHORT).show();

if (c1.isChecked() == false && c2.isChecked() == true && c3.isChecked() == true)

Toast.makeText(MainActivity.this, "Coffee" + c + "\n Burger" + b + "\n Total = " + sum, Toast.LENGTH_SHORT).show();

if (c1.isChecked() == true && c2.isChecked() == false && c3.isChecked() == true)

sum = p + b;

Toast.makeText(MainActivity.this, "Pizza" + p + "\n Burger" + b + "\n Total = " + sum, Toast.LENGTH_SHORT).show();

```
if (c1.isChecked() == true && c2.isChecked() == true) {
    c3.isChecked() == true
}
sum = p+c+h
Total.setText("Main Activity, this " + p + " + " + c + " + " + h + " = " + sum);
Toast.LENGTH_SHORT).show();
}
}
}
}
```

(Space for answers)

```

Q 1) <time picker
android:id="@+id/time_picker"
android:layout_width="wrap_content"
android:layout_height="32dp"
android:layout_x="32dp"
android:layout_y="102dp"
    
```

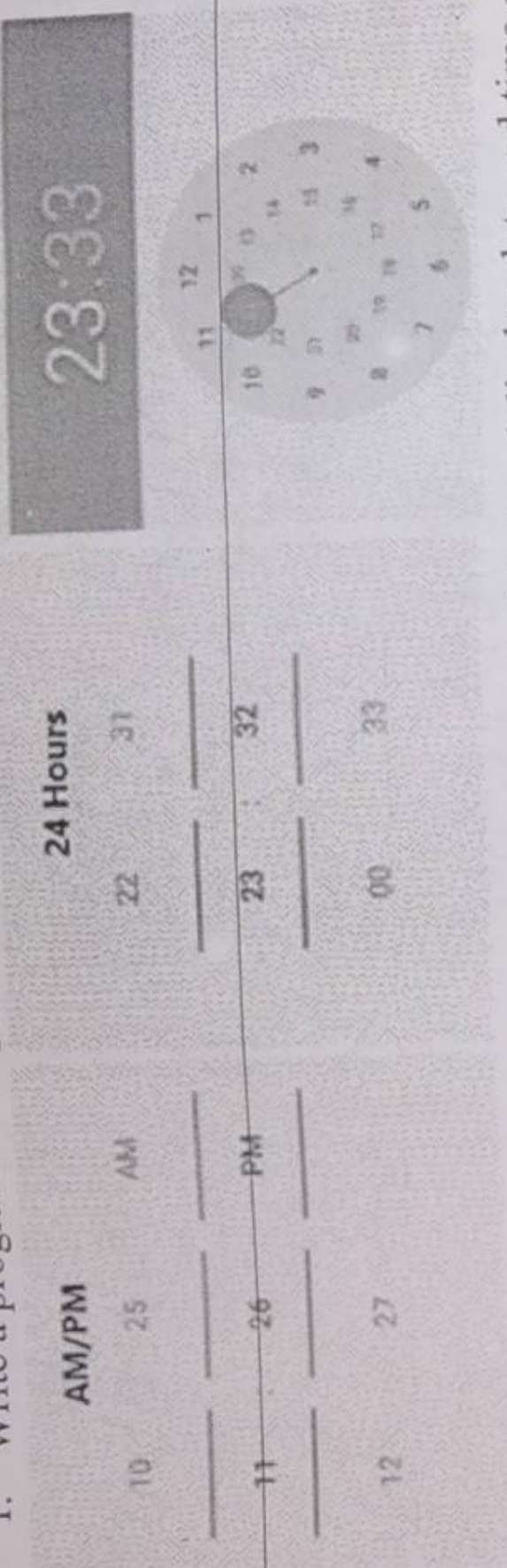
- Q 2) → 1) `is24HourView()`
 This method returns true if this is in 24 hour view else false
- 2) `isEnabled()`
 This method returns the enabled status for this view.
- 3) `setCurrentHour(Integer currentHour)`
 This method sets the current hour
- 4) `setCurrentMinute(Integer currentMinute)`
 This method sets the current minute
- 5) `setEnabled(boolean enabled)`
 This method sets enabled state of this view.

- Q 3) → 1) `isSpinnerShown(boolean shown)`
 To set whether the spinner is shown or not.
- 2) `getDayOfMonth()`
 To get the selected day of month
- 3) `getMonth()`
 To get the selected month
- 4) `getYear()`
 To get the selected year
- 5) `getFirstDayOfWeek()`
 To get the first day of week.

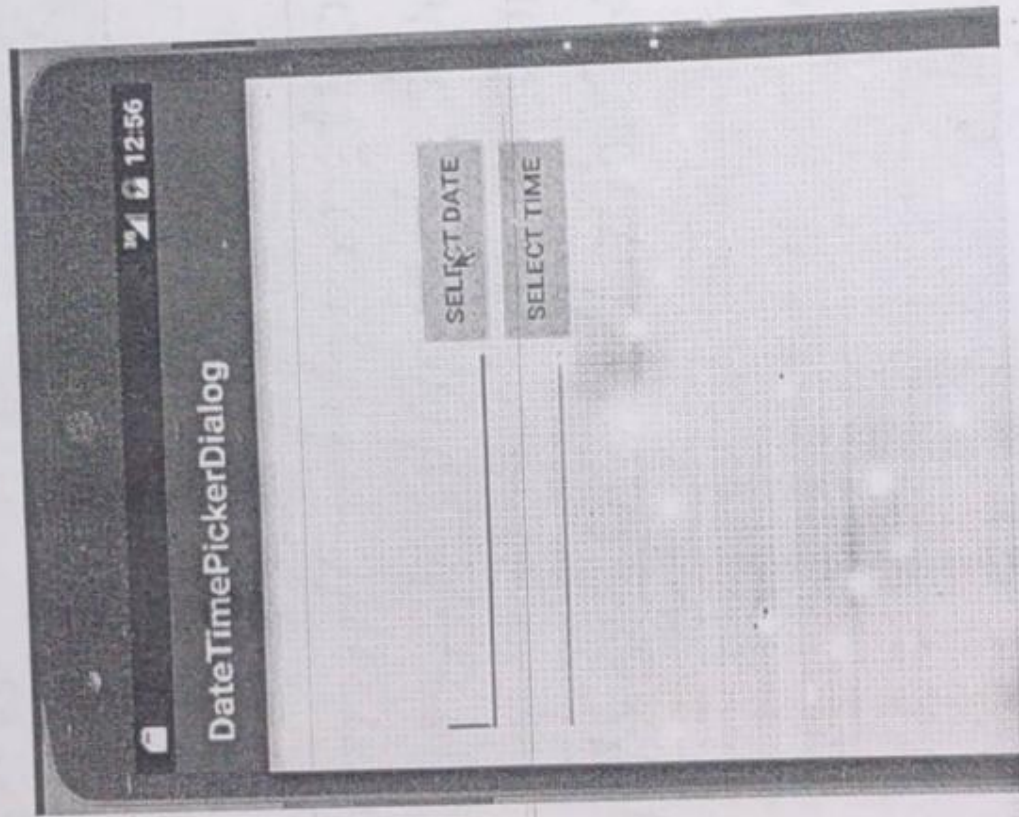
X. Exercise

(Use blank space provide for answers or attached more pages if needed)

1. Write a program to display following output. Use TimePicker with Spinnermode.



2. Write a program to display following output. Select and display date and time on click of "select date", "select time" buttons respectively.



(Space for answers)

```

1) <LinearLayout xmlns:android="http://schemas
    , android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.shruti.exp16.1
    MainActivity"
    tools:ignore="MergeRootFrame"/>
    
```

```

<TextView
    android:id="@+id/textview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    tools:context="com.stash"
    android:text="textview"/>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"/>
<TimePicker
    android:id="@+id/timepicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    </LinearLayout>
public class MainActivity extends
    AppCompatActivity {
    TextView t1;
    Button b1;
    TimePicker tPl;
    b1=(TextView) findViewById(R.id.textview);
    b1=(Button) findViewById(R.id.button);
    tPl=(TimePicker) findViewById(R.id.timepicker);
    onClickListener() {
        public void onClick(View arg0) {
            t1.setText(getCurrentTime());
        }
    };
    private CharSequence getCurrentTime() {
        String T = "Current Time = " + tPl;
        return getCurrentHour() + " " + tPl;
    }
}

```



```
getCurrencyMinute();  
return();  
}
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1. Logic Formation		10%
2. Debugging ability		15%
3. Follow ethical practices		5%
Product related (15 Marks)		70%
4. Interactive GUI		20%
5. Answer to Practical related questions		20%
6. Expected Output		20%
7. Timely Submission		10%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained		Dated signature of Teacher
Process Related(10)	Product Related(15)	
Total (25)		

```

2) -> activity main.xml:-
<LinearLayout xmlns:android="http://schemas.android
.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/container"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_orientation="vertical"
tools:context="com.shree.exp16"
MainActivity
tools:ignore="MergeRootFrame"
<TextView
android:id="@+id/textview"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="TextView" />
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_text="Button" />
<DatePicker
android:id="@+id/datePicker"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
</LinearLayout>
MainActivity.java
Package com.shree.exp16.2;

```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
import android.os.Build;

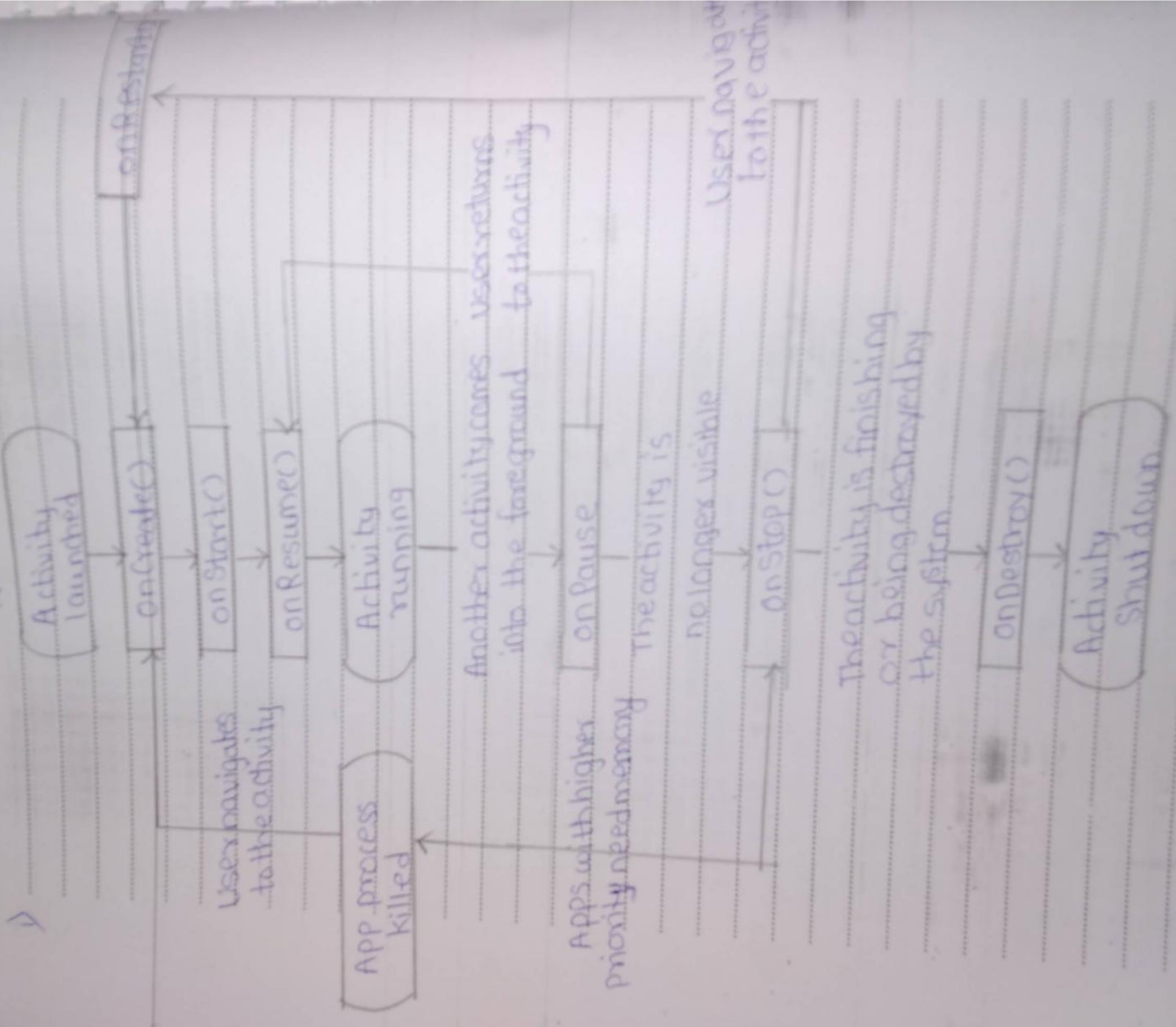
public class MainActivity extends
    Activity {
    TextView t1;
    Button b1;
    DatePicker d1;

    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main);
        t1 = (TextView) findViewById(R.id.textView);
        b1 = (Button) findViewById(R.id.button);
        d1 = (DatePicker) findViewById(R.id.datepicker);
        d1.setCalendarViewShown(false);
        t1.setText(getCurrentDate());
        b1.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View arg0) {
                t1.setText(getCurrentDate());
            }
        });
    }
}
```

```
private char Sequence get Current Date
```

```
{  
    String Builder sb = new String Builder  
    sb.append(d1.getMonth() + "/" + "  
    sb.append(d1.getDayOfMonth() + "/" + "  
    sb.append(d1.getYear());  
    return sb.toString();  
}
```

(Space for answers)



2) →

res/layout/activity_main.xml

res/layout/mainActivity.java

3) →

1) onStop() :-

The activity is completely hidden & not visible to the user

2) onDestroy() :-

Activity is destroyed & removed from the memory.

3) onPause() :-

Activity is destroyed & removed partially, observed by another activity, another activity that's in the foreground is semi-transparent.

4) onResume() :-

The activity is completely hidden & not visible to the user.

X. Exercise

(Use blank space provide for answers or attached more pages if needed)

1. Write a program to create a Hello World Activity using all lifecycle methods to display messages using Log.d.

(Space for answers)

```

1) →
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res-auto"
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="example.javatpoint.com.activity
lifecycle.MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toTopOf="parent"/>
</android.support
package example.javatpoint.com.activity.lifecycle;
import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
public class MainActivity extends Activity {
    @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();  
    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 invoked");  
}  
@Override  
protected void onStop() {  
    super.onStop();  
    Log.d("Lifecycle", "onStop invoked");  
}  
@Override  
protected void onRestart() {  
    super.onRestart();  
    Log.d("Lifecycle", "onRestart invoked");  
}  
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    Log.d("Lifecycle", "onDestroy invoked");  
}  
}
```


IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List different methods used in Intent.
2. Write an intent to display the phone dialer with the given number filled in.

(Space for answers)

- 1) addCategory()
- 2) addFlags()
- 3) clone()
- 4) cloneFilter()
- 5) describeContents()

2) →
Intent i = new Intent(Intent.ACTION_VIEW);
Uri.parse("tel: 9834414901")

X. Exercise

(Solve any one of the following. Use blank space provide for answers or attached more pages if needed)

1. Write a program to create a text field and a button "Navigate". When you enter "www.google.com" and press navigate button it should open google page.
2. Write a program to create button "Start Dialer". When u click on this button it should open the phone dialer.
3. Write a program to create two screens. First screen will take one number input from user. After click on Factorial button, second screen will open and it should display factorial of the same number. Also specify which type of intent you will use in this case.

(Space for answers)

```

1) Activity-main.xml .....
<AbsoluteLayout xmlns:android="http://schemas
  android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/container"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  textContext="com.smriti.expl7.MainActivity"
  tools:ignore="MergeRootFrame" >
  <EditText
  android:id="@+id/editText"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:ems="10" />
  <Button
  android:id="@+id/button1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="All" />
</AbsoluteLayout>
MainActivity.java
import android.app.Activity;
import android.app.ActionBar;

```

```

public class MainActivity extends Activity
{
    EditText et;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et = (EditText) findViewById(R.id.editText);
        et.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent i = new Intent(Intent.ACTION_DIAL,
                Uri.parse("tel:95601574179"));
                startActivity(i);
            }
        });
    }
}

```

```

2) → Activity_main.xml -
<LinearLayout xmlns:android="http://schemas.
android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/container"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:ignore="orientation"
<Button
android:id="@+id/button"
android:layout_width="wrap_content"

```

```
android:layout_height = "wrap_content"
android:layout:text = "search" />
<EditText
    android:id = "@id/edittext1"
    android:layout_width = "match_parent"
    android:layout_height = "match_parent"
    android:ems = "10"
    <request focus />
</EditText>
<Button
    android:id = "@id/button1"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "factorial" />
</LinearLayout>
```

MainActivity.java -

```
package com.javatpoint.expl8.3;
import android.app.Activity;
import android.app.ActivityBar;
import android.app.Fragment;
import android.app.Content.Intent;
import android.view.Menu;
import android.os.Build;
public class MainActivity extends
```

Activity {

Button b1;

EditText e1;

@Override

protected void onCreate (Bundle savedInstanceState

super.onCreate (saveInstanceState);

```

setContentView(R.layout.activity_main);
b1 = (Button)findViewById(R.id.button);
e1 = (EditText)findViewById(R.id.editText);
b1.setOnClickListener(new View.OnClickListener()

```

```

}
@Override

```

```

public void onClick(View arg0)

```

```

{
    Intent i = new Intent(MainActivity.this,
    sec_activity.class);
    i.putExtra("username", e1.getText().toString());
    startActivity(i);
}
}
}

```

```

3) <?xml xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.shruti.exp.MainActivity"
    tools:ignore="MergeRootFrame" >
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView" />
    </frameLayout>

```

```

Second-act.java :-

```

```

import android.app.Activity;

```

```

import android.support.design.widget.AppBarLayout;

```

```
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
public class MainActivity extends
    Activity {
    TextView t1;
    String s;
    int d, fact = 1, n = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second_main);
        t1 = (TextView) findViewById(R.id.textView);
        Intent i = getIntent();
        int i = Integer.parseInt(i.getStringExtra(
            getString("username")));
        while (a > 0)
        {
            fact = fact * a;
            a--;
        }
        t1.setText("factorial: " + Integer.toString(fact));
    }
}
```

VIII. Resources required (Additional)

Sr. No.	Instrument / Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical Related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write in detail which methods are needed to implement Content Provider class.
2. Explain different parts of an URI in android application. Also write the format of URI.
3. Write steps to create a content provider in android applications.

(Space for answers)

1) `onCreate()` - This method is called when the provider is started.
 2) `query()` - This method receives a request from a client. The result is returned as a cursor object.
 3) `insert()` - This method inserts a new record into content provider.
 4) `delete()` - This method deletes an existing record from the content provider.
 5) `update()` - This method updates an existing record from the content provider.
 6) `getType()` - This method returns the MIME type of data at the given URI.

URI -> The scheme, subdomain, top-level domain, second-level domain and subdirectory.

Scheme component: it is a non-empty component followed by a colon (:). Scheme contains a sequence of characters starting with a letter and followed by any combination of digits, letters, period (.), hyphen (-), or plus (+).

- 3) → 1) Create a class that extends Content Provider
- 2) Create a contract class
- 3) Create the UriMatcher definition
- 4) Implement the onCreate() method
- 5) Implement the getType() method
- 6) Implement the CRUD methods
- 7) Add the content provider to your AndroidManifest.xml

Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to create your own content provider to insert and access data in android application.

```

1) → package com.Example.MyApplication;
import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.ContentValues;
import android.database.Cursor;
import android.content.CursorLoader;
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void onClickAddName(View view) {
ContentValues values = new ContentValues();
values.put("StudentProvider.NAME",
((EditText)findViewById(R.id.editText2)).getText()
.toString());
Uri uri = getContentResolver().insert(StudentProvider
.CONTENT_URI, values);
Toast.makeText(getApplicationContext(),
uri.toString() + Toast.LENGTH_LONG).show(); }
    
```



```

public void onClickRetrieveStudents(View view)
{
    String URL = "content://com.example.myapplication.
    students.provider";
    Uri student = Uri.parse(URL);
    Cursor c = managedQuery(students, null,
    null, null, "name");
    if (c.moveToFirst()) {
    do {
        Toast.makeText(this, c.getString(c.getColumnIndex
        Index (students.provider_ID)) +
        ", " + c.getString(c.getColumnIndex (students
        provider_name)) +
        ", " + c.getString(c.getColumnIndex (students
        provider_grade)),
        Toast.LENGTH_SHORT).show();
    } while (c.moveToNext());
    }
}
}
}
}

```

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

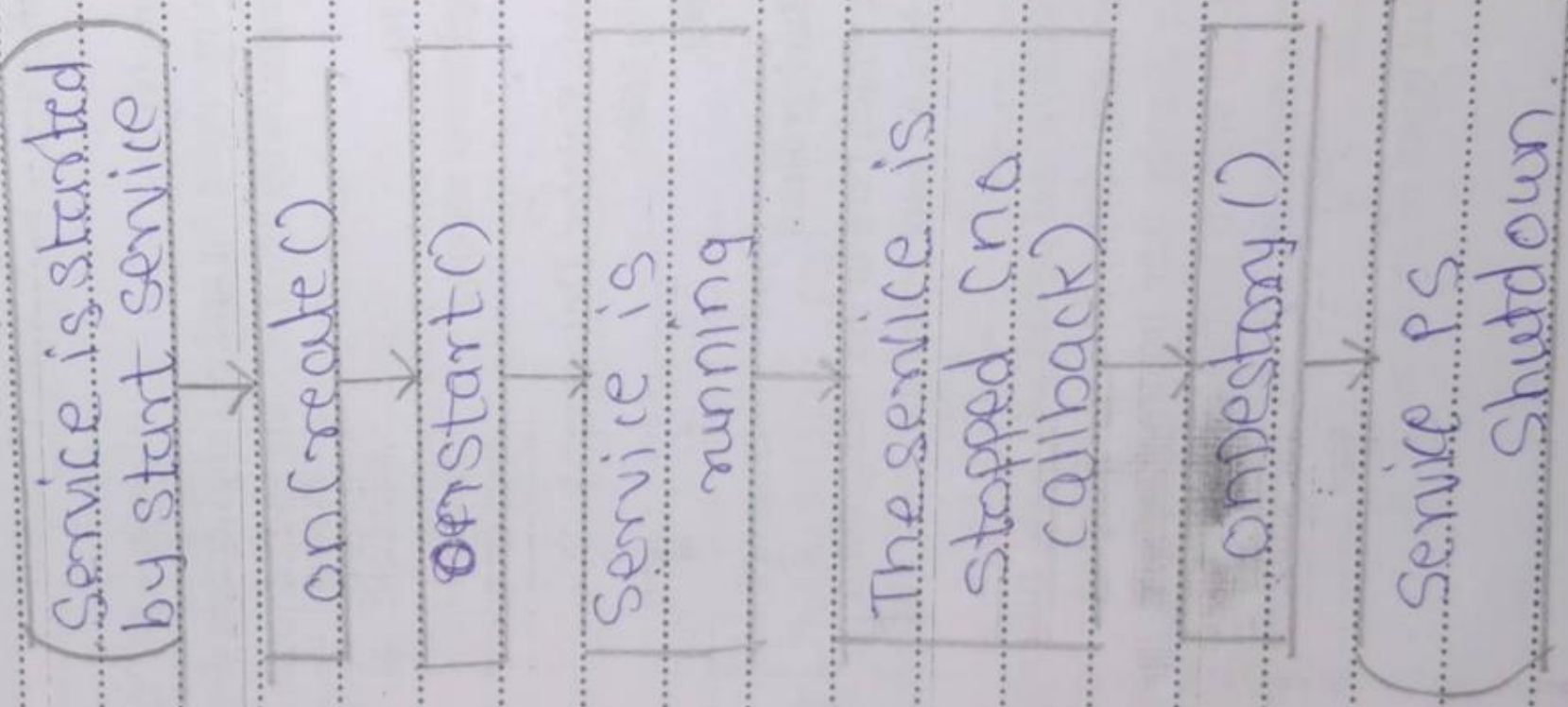
IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Draw the lifecycle of service.
2. Differentiate between bounded service and unbounded service.
3. Describe startService() and bindService() methods.

(Space for answers)

1) →



2) →

Bounded service	Unbounded service
1) Bounded Service is used to perform background task in bound with another component	1) Unbounded service is used to perform long repetitive task
2) Bounded Service gets starts by calling bindService()	2) Unbound service gets starts by calling startService()
3) bounded service is unbind or destroyed by calling unbindService()	3) Unbound Service is stopped or destroyed explicitly by calling stopService()
4) bound service depends on the component in which it started	4) Unbound service is independent of the component in which it is started.

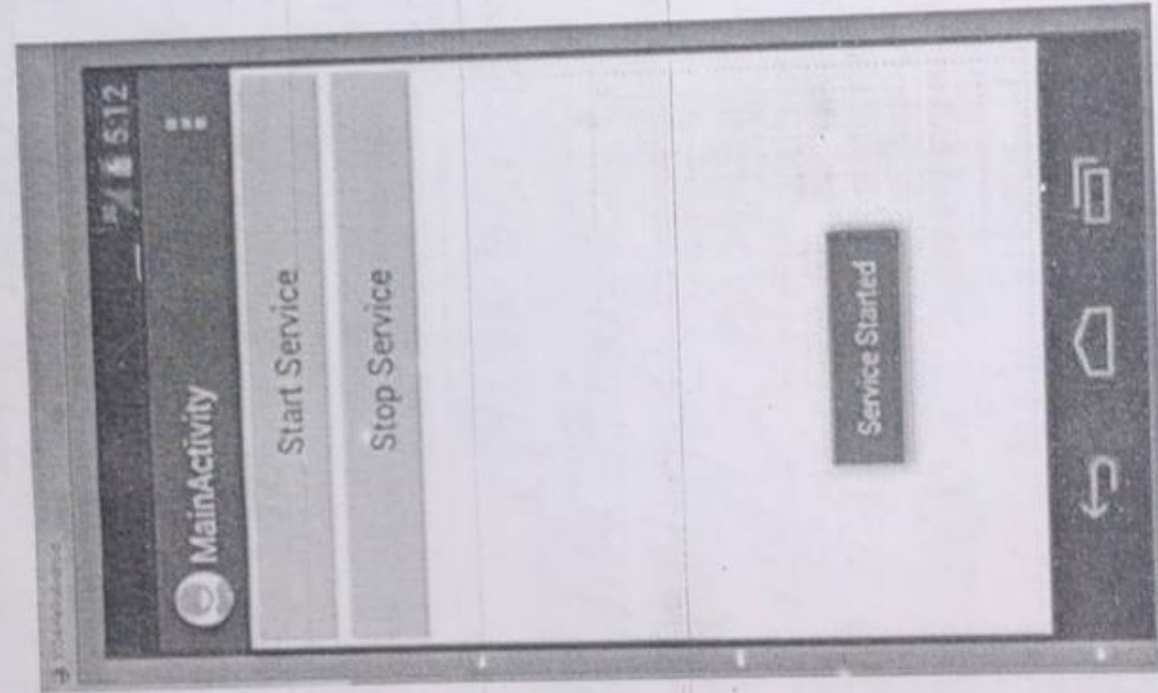
- 3) → startService() - A service is started when an application component, such as an activity, starts it by calling startService()
- bindService() - A service is bound when an application component binds to it by calling bindService().

.....
.....
.....
.....

X. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to start a Wi-Fi using service.
2. Write a program to display the following output.



(Space for answers)

```

<RelativeLayout xmlns:android="http://schemas.android.com/tools"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context="MainActivity">
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap-content"
        android:layout_height="wrap-content"
        android:layout_alignParentLeft="true"

```

```

android:layout_alignParentTop = "true"
android:layout_marginLeft = "7.6dp"
android:layout_marginTop = "67dp"
android:text = "Enable Wifi" />
<Button

```

```

    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button1"
    android:layout_below="@+id/button1"
    android:layout_marginTop="44dp"
    android:text="Disable Wifi" />
</RelativeLayout>

```

```

MainActivity.java
package com.example.wifi;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainActivity extends Activity {
    Button enableButton, disableButton;
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    enableButton = (Button) findViewById(R.id.button1);
    disableButton = (Button) findViewById(R.id.button2);
    enableButton.setOnClickListener(new OnClickListener() {

```

```
wifiManager wifi = (WifiManager) getSystemService  
wifi.setWifiEnabled(false);  
} };
```

```
@Override
```

```
public boolean onCreateOptionsMenu (Menu menu) {  
getMenuInflater().inflate (R.menu.activity_main,  
return true;  
}
```

```
<uses -
```

```
permission android:name = "android.permission.  
ACCESS_WIFI_STATE"/>
```

```
<uses -
```

```
permission android:name = "android.permission  
.INTERNET"/>
```

```
<uses -
```

```
permission android:name = "android.permission.  
CHANGE_WIFI_STATE"/>
```

```
> -> <LinearLayout xmlns:android = "http://schemas  
.android.com/apk/res/android"  
xmlns:tools = "http://schemas.android.com/tools"  
android:id = "@+id/container"/>
```

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context="com.shruti.Exp20-1:MainActivity"
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="Start Service">
</Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="Stop Service">
</LinearLayout>
MainActivity.java -
import android.app.Activity;
public class MainActivity extends Activity
{
    @Override
    Button b1, b2;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        b1 = (Button) findViewById(R.id.button)
        b2 = (Button) findViewById(R.id.button)
        b1.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View arg1)

```

```
{
Intent i = new Intent (MainActivity.this, SecondActivity.class);
startService(i);
}
b2.setOnClickListener (new View.OnClickListener()
{
@Override
public void onClick (View arg)
{
Intent i = new Intent (MainActivity.this, SecondActivity.class);
startService(i);
}
});
}
}

SecondActivity.java
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
public class SecondActivity extends Service {
@Override
public IBinder onBind (Intent arg)
{
return null;
}
}
```



```

{ public void onClick(View v) {
    WifiManager wifi = (WifiManager) getSystemService(
        Context.WIFI_SERVICE);
    wifi.setWifiEnabled(true);
} }
disableButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {

```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1. Logic Formation		10%
2. Debugging ability		15%
3. Follow ethical practices		5%
Product related (15 Marks)		70%
4. Interactive GUI		20%
5. Answer to Practical related questions		20%
6. Expected Output		20%
7. Timely Submission		10%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained		Dated signature of Teacher
Process Related (10)	Product Related (15)	



VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

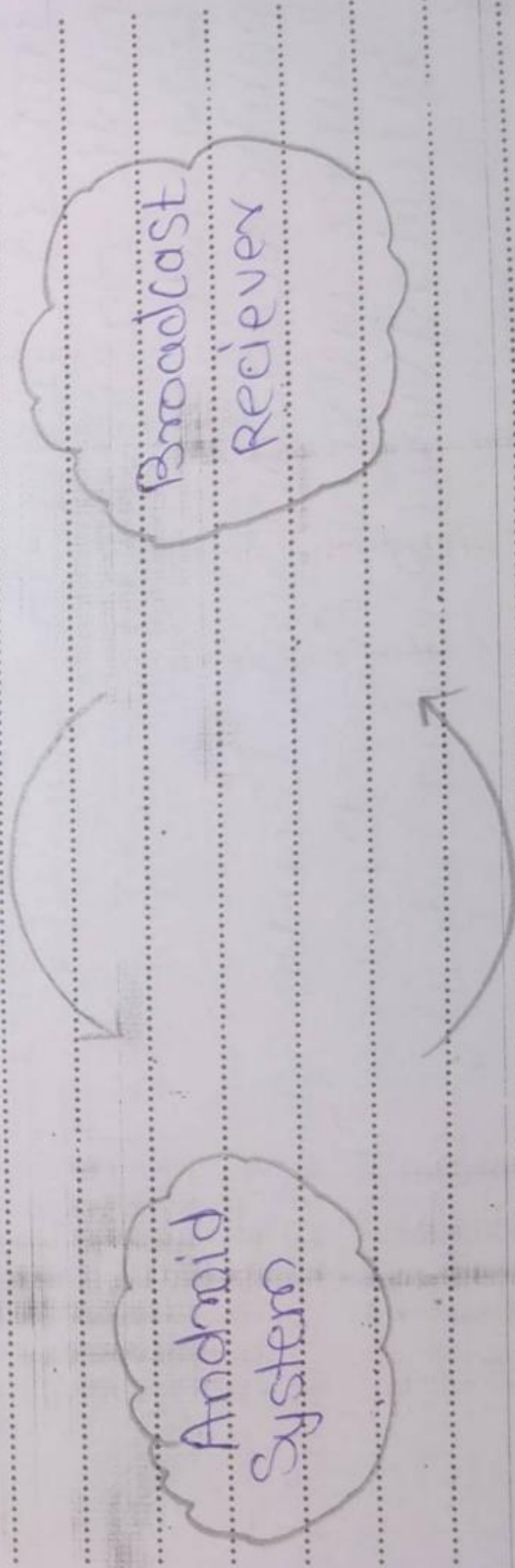
Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Differentiated between Activity Intent and Broadcasting Intent.
2. Draw Broadcast Receivers Lifecycle.
3. List the System Events related to Broadcast Receivers.

(Space for answers)

e) → Broadcast Receivers Lifecycle

Registers for Intents to observe



Gets Notification when Intents occur

- 3) i) Intent Action Boot COMPLETED
- ii) Intent Action Power CONNECTED
- iii) Intent Action Power DISCONNECTED
- iv) Intent Action Battery low
- v) Intent Action REBOOT
- vi) Intent Action DATE CHANGED

1) An application component for displaying a user interface. The activity class is where all user interactions are handled. An activity specifies a layout to represent it on screen.

An intent is a system message. It can be broadcast around the system to notify other application of an event or it can be used to request that the system display a new activity.

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to demonstrate all the system broadcast messages.

(Space for answers)

```
package com.example.tutorialspoint7;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```

public void broadcastIntent (view.view) {
    Intent intent = new Intent();
    Intent.setAction ("com.tutorialspoint.
    CUSTOM_INTENT).sendBroadcast(intent);
}
}

package com.example.tutorialspoint.myapplication

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, "Intent Detected", Toast.
        LENGTH_LONG);
    }
}

    Butterknife.inject(this);
    receiverReceiver (receiver, intentFilter);
}
@Override
protected void onDestroy () {
    super.onDestroy();
    unregisterReceiver (receiver);
}
    @OnClick (R.id.button)
    void someMethod () {
        Intent intent = new Intent ("com.journaldev.broadcast
        s.receiver SOME_ACTION");
        sendBroadcast (intent);
    }
}
}

```

Once that sensor is declared, you need to register its listener and override two methods which are `onAccuracyChanged` and `onSensorChanged`. Its syntax is as follows:

```
sMgr.registerListener(this, lightSensorManager.SENSOR_DELAY_NORMAL);
public void onAccuracyChanged(Sensor sensor, int accuracy)
{}
public void onSensorChanged(SensorEvent event) {}
```

Methods:

1. `getDefaultSensor(int type)` :- This method get the default sensor for a given type. Explain methods
2. `getOrientation(float[] R, float[] values)` :- This method returns a description of the current primary clip on the clipboard but not a copy of its data.
3. `getInclination(float[] I)` :- This method computes the geomagnetic inclination angle in radians from the inclination matrix.
4. `registerListener(SensorListener listener, int sensors, int rate)` :-This method registers a listener for the sensor
5. `unregisterListener(SensorEventListener listener, Sensor sensor)` :-This method unregisters a listener for the sensors with which it is registered.
6. `getOrientation(float[] R, float[] values)` :-This method computes the device's orientation based on the rotation matrix.
7. `getAltitude(float p0, float p)` :-This method computes the Altitude in meters from the atmospheric pressure and the pressure at sea-level.

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List the best practices for accessing and using sensors.
2. Differentiate between Sensor Class and Sensor Manager Class.

(Space for answers)

i) → i) unregisters sensor listeners
 ii) Don't block the `onSensorChanged()` method
 iii) Avoid using deprecated method on sensor types
 iv) Verify sensors before you see them

1) Choose sensor delays carefully

2) Sensor class -
This class is used to create an instance of a specific sensor

Sensor Manager class -
This is used to get access to various sensors present in the device to use it according to need

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to change the background color when device is shuffled.
2. Write a program to display the list of sensors supported by the mobile device.

(Space for answers)

1)

```
activity_main.xml
<RelativeLayout xmlns:android="http://schemas.android
xmlns:tools="http://schemas.android.com/tools
android:layout_width="match_parent"
android:layout_height="match_parent"
```

```
tools: content=".MainActivity">
<TextView
    android:id="@+id/textView"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    android:text="shake to switch color"/>
</RelativeLayout>
```

MainActivity.java

```
import android.app.Activity;
import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends Activity
    implements SensorEventListener {
    private SensorManager sensorManager;
    private boolean isColor = false;
    private View view;
    private long lastUpdate;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        view = findViewById(R.id.textView);
        view.setBackgroundColor(Color.GREEN);
        sensorManager = (SensorManager) getSystemService(
            SENSOR_SERVICE);
        lastUpdate = System.currentTimeMillis();
    }
    @Override
```

```

public void onAccuracyChange(Sensor sensor,
int accuracy) {}
@Override
public void onSensorChanged(SensorEvent event)
{ if (event.sensor.getType() == Sensor.TYPE_
ACCELEROMETER) {
getAccelerometer(event);
}
}
private void getAccelerometer(SensorEvent event)
{
}

```

e)

```

-> <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"
android:transitionGroup="true">
<TextView android:text="Motion Sensor"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
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="Position Sensor"  
android:id="@+id/textView"  
android:layout_below="@+id/textView"  
android:layout_centerHorizontal="true"  
android:textColor="#ff7aff24"  
android:textSize="135dp" />
```

```
<TextView  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Environmental Sensor"  
android:id="@+id/textView2"  
android:layout_below="@+id/imageview"  
android:layout_alignParentBottom="true"  
android:layout_alignParentRight="true"  
android:layout_alignParentEnd="true"  
android:layout_alignParentLeft="true"  
android:layout_alignParentStart="true" />  
</RelativeLayout>
```

- 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.

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone/ Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List all the methods related to camera class.
2. Explain the method that is used to detect the face.

(Space for answers)

Q1) 1) ACTION_IMAGE_CAPTURE_SECURE
 2) ACTION_VIDEO_CAPTURE
 3) EXTRA_SCREEN_ORIENTATION
 4) EXTRA_FULL_SCREEN

Q2) → startface detection
 public void startfaceDetection() {
 // It starts face detection the camera will
 // notify face detection listener of detected
 // faces in preview frame

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Solve any one of the following. Use blank space for answers or attach more pages if needed)

1. Write a program to capture an image and display it using image view.
2. Write a program to record a video using various camera methods.

(Space for answers)

```
→ <?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://
schemas.android.com/apk/res/android
xmlns:tools="http://schemas.android.com/tools"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Open camera"
<ImageView
android:id="@+id/img1"
android:layout_width="match_parent"
```

```

android:layout_height="351dp"
tools:serccompat="@tools
sample/avatars [6]" />
</LinearLayout>
import android.content.Intent;
import android.provider.Media;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends Activity
AppCompatActivity {
    Button b1;
    ImageView iv1;
    private static final int pic = 123
    @Override
    protected void onCreate(Bundle savedInstanceState)
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity);
    b1 = (Button) findViewById(R.id.button)
    iv1 = (ImageView) findViewById(R.id.imageView);
    b1.setOnClickListener(new OnClickListener()
    @Override
    public void onClick(View v)
    Intent i = new Intent (MediaStore.ACTION_
    IMAGE_CAPTURE_SECURE);
    }
    }
}

```

```

2) package com.javacodegeeks
import java.io.IOException;
import android.app.Activity;
import android.content.Content
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
    
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

```
public class AndroidVideoCaptureExample extends
Activity {
    private Camera mCamera;
    private CameraPreview mPreview;
    private MediaRecorder mediaRecorder;
    private Button capture, switchCamera;
    private Context myContext;
    private LinearLayout cameraPreview;
    private boolean cameraFront = false;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getWindow().addFlags(WindowManager.LayoutParams.
        FLAG_KEEP_SCREEN_ON);
        myContext = this;
        initialize();
    }
    private int findFrontfacingCamera() {
        int cameraId = -1;
        int numberOfCameras = Camera.getNumberOfCameras();
        for (int i = 0; i < numberOfCameras; i++) {
            CameraInfo info = new CameraInfo();
            camera.getCameraInfo(i, info);
            if (info.facing == CameraInfo.CAMERA_FACING_FRONT)
                cameraId = i;
            cameraFront = true;
            break;
        }
    }
    return cameraId;
}
```

```
private int findBackFacingCamera() {
    int cameraId = -1;
    int numberOfCameras = Camera.getNumberOfCameras();
    for(int i = 0; i < numberOfCameras; i++) {
        CameraInfo info = new CameraInfo();
        Camera.getCameraInfo(i, info);
        if(info.facing == CameraInfo.CAMERA_FACING_BACK) {
            cameraId = i;
            cameraFront = false;
            break;
        }
    }
    return cameraId;
}

public void onResume() {
    super.onResume();
    if(!hasCamera(myContext)) { Toast toast =
        Toast.makeText(myContext, "Sorry, your phone
        does not have a camera!",
        try {
            mediaRecorder.prepare();
        } catch (IllegalStateException e) {
            releaseMediaRecorder();
            return false;
        } catch (IOException e) {
            releaseMediaRecorder();
            return false;
        }
    }
    return true;
}
```

```
}  
private void releaseCamera () {  
    //stop and release camera  
    if (mCamera != null) {  
        mCamera.release ();  
        mCamera = null ;  
    }  
}
```



Once you enable the Bluetooth, you can get a list of paired devices by calling `getBondedDevices()` method. It returns a set of Bluetooth devices. Its syntax is.

```
private  
Set<BluetoothDevice>pairedDevices;  
pairedDevices =  
BA.getBondedDevices();
```

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Name the methods which are used to enable and disable Bluetooth adapter.
2. Explain the purpose of `ACTION_REQUEST_DISCOVERABLE` Constant.
3. List the uses of `setName(String name)` method.

(Space for answers)

Q1)

→ 1) To enable bluetooth adapter -
`BluetoothAdapter.ACTION_REQUEST_ENABLE`

2) To disable bluetooth adapter
`BluetoothAdapter.disable()`

Q2)

→ `ACTION_REQUEST_DISCOVERABLE` use to turn on discovering of bluetooth.

Q3)

→ `setName(StringName)`
This method changes the bluetooth name

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to turn on, get visible, list devices and turnoff Bluetooth with the help of following GUI.



(Space for answers)

```
→ package com.shruti.exp2221;  
import android.content.Intent;  
import android.bluetooth.BluetoothDevice;  
import android.view.View;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.widget.Button;  
import android.widget.Toast;
```

```

import java.util.set;
public class MainActivity extends AppCompatActivity {
    Button b1, b2, b3, b4;
    BluetoothAdapter ba;
    private Set<BluetoothDevice> name;
    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ba = BluetoothAdapter.getDefaultAdapter();
        b1 = (Button) findViewById(R.id.button);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick (View v) {
                Intent a = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(a, 0);
            }
        });
        tools:context=".MainActivity">
        <TextView
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_text="Bluetooth"/>
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Turn on"/>
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Get visible"/>

```

```

<Button
  android:id="@id/button3"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="List Device"/>

```

```

<Button
  android:id="@id/button4"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Turn off"/>
</LinearLayout>

```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

Mobile Application Development (22617) : This method sets the duration of an animation.

2. setDuration(long duration) : This method gets the duration which is set by above method.
3. getDuration() : This method gets the duration.
4. end() : This method ends the animation.
5. cancel() : This method cancels the animation.

VIII. Resources required (Additional)

Sr. No.	Instrument / Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write the steps to perform Tween Animation.
2. Explain the use of from XScale and from YScale method in detail.

(Space for answers)

Q1) →

step 1: Right click on res folder and choose new folder option then type the name of the folder in the dialog box appears and press Enter
 step 2: Now right click the anim folder and select android.xml file from the options.
 step 3: A dialog box will appear. Type the name of file and press Enter.

Q2) →

from X -

Horizontal scaling factor to apply at the start of the animation

from Y -

Vertical scaling factor to apply at the start of the animation

.....

.....

.....

.....

.....

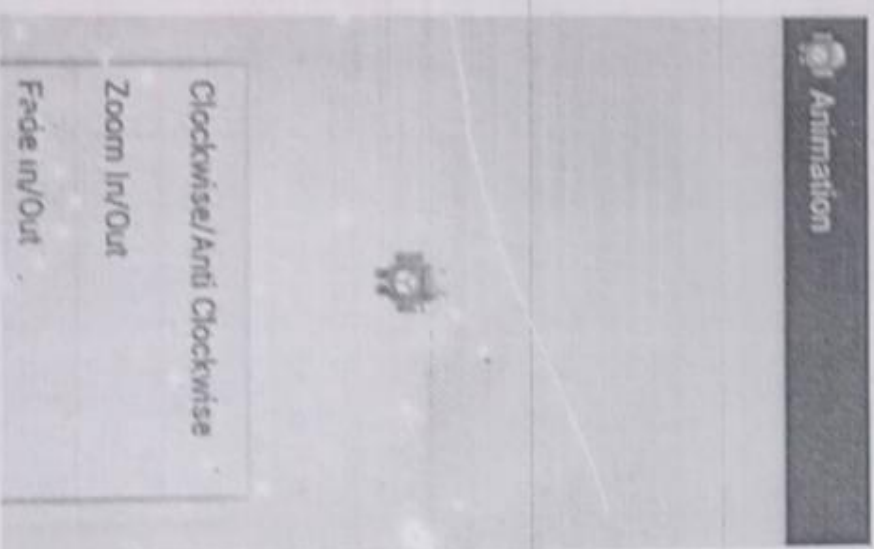
.....

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to rotate the image in clockwise/anticlockwise, Zoom IN/Zoom OUT, Fade IN/Fade OUT by using the following GUI.



(Space for answers)

```
→ package com.javacodegeeks.android.animations;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.Animation;
```

```

import androidx.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity {
    private Button moveIntent;
    private Button rotate;
    private TextView myText;
    private ImageView myImage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myText = (TextView) findViewById(R.id.fade);
        myImage = (ImageView) findViewById(R.id.image);
        fade = (Button) findViewById(R.id.fade);
        fade.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View V) {
                fadeAnimation(V);
            }
        });
        zoom = (Button) findViewById(R.id.zoom);
        zoom.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View V) {
                zoomAnimation(V);
            }
        });
        moveIntent = (Button) findViewById(R.id.move);
        moveIntent.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View V) {

```

```

moveAnimation(v);
});
rotate=(Button)findViewById(R.id.rotate);
rotate.setOnClickListener(new OnClickListener()
@Override
public void onClick(View v) {
rotateAnimation(v);
}
}
    
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators	Weightage
Process related (10 Marks)	30%
1. Logic Formation	10%
2. Debugging ability	15%
3. Follow ethical practices	5%
Product related (15 Marks)	70%
4. Interactive GUI	20%
5. Answer to Practical related questions	20%
6. Expected Output	10%
7. Timely Submission	10%
Total (25 Marks)	100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related (10)	Product Related (15)	Total (25)	


```

    });
    private void fadeAnimation (View view) {
        Animation animation =
            AnimationUtils.loadAnimation (getApplication
            context(), R.anim.fade);
        myText.startAnimation (animation);
    }
    private void zoomAnimation (View view) {
        Animation animation = AnimationUtils.load
        Animation (getApplication context(), R.anim.zoom);
        myImage.startAnimation (animation);
    }
    private void moveAnimation (View view) {
        my {
            Intent intent = new Intent (this, MoveActivity.class);
            startActivity (intent);
        } catch (Exception) {
            Toast.makeText (getApplication context(), "Error
            with the Intent",
            Toast.LENGTH_SHORT).show();
            e.printStackTrace();
        }
    }
}

```

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List the basic methods used in an android AsyncTask class.
2. Differentiate between AsyncTask and Services.
3. Name the method used, if a process takes a long time to do its work?

(Space for answers)

Q1)

- 1) doInBackground()
- 2) onPreExecute()
- 3) onPostExecute()
- 4) onProgressUpdate()

Q2)

→ AsyncTask are designed for once off time-consuming tasks that cannot be run on UI thread.

Services are designed to be continually running in background.

Q3)

→ onStart() onResume() method used if a process takes a long time to do its work.

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to insert data in SQLite database using AsyncTask

(Space for answers)

```
<RelativeLayout
xmlns:android="https://schemas.android.com/
apk/res/android"
xmlns:tools="https://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
android:id="@+id/tv_time"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textSize="10pt"
android:textColor="#444444"
android:layout_alignParentLeft="true"
android:layout_marginRight="9dip"
android:layout_marginTop="20dip"
android:layout_marginLeft="10dip"
android:text="Sleep time in seconds:"/>
<EditText
android:id="@+id/in_time"
android:layout_width="150dip"
android:layout_height="wrap_content"
android:background="@android:drawable/editbox_
background"
android:layout_toRightOf="@id/tv_time"
android:layout_alignTop="@id/tv_time"
android:inputType="number"
<Button
android:id="@+id/btn_run"
```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Run Async task"
android:layout_below="@+id/in_time"
android:layout_centerHorizontal="true"
android:layout_marginTop="64dp" />

```

```

<TextView

```

```

android:id="@+id/tv_result"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textSize="7pt"
android:layout_below="@+id/btn_run"
android:layout_centerHorizontal="true" />

```

```

</RelativeLayout>

```

```

public class MainActivity extends AppCompatActivity {

```

```

    private Button button;

```

```

    private EditText time;

```

```

    private TextView finalResult;

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState)

```

```

    {
        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_main);

```

```

        time = (EditText) findViewById(R.id.in_time);

```

```

        button = (Button) findViewById(R.id.btn_run);

```

```

        finalResult = (TextView) findViewById(R.id.tv_result);

```

```

        button.setOnClickListener(new View.OnClickListener()

```

```

        {

```

```

            public void onClick(View v) {

```

```

                AsyncTaskRunner runner = new AsyncTaskRunner();

```

```

                String sleepTime = time.getText().toString();

```

```

                runner.execute(sleepTime);

```

```

            }

```

```

        });

```

```

    }

```

```
private class AsyncTaskRunner extends AsyncTask<String, String, String> {
    private String resp;
    ProgressDialog progressDialog;
    @Override
    protected String doInBackground(String... params) {
        publishProgress("sleeping...");
        try {
            int time = Integer.parseInt(params[0])*1000;
            Thread.sleep(time); resp = "slept for " + params[0] + "
seconds";
        } catch (InterruptedException e) {
            e.printStackTrace();
            resp = e.getMessage();
        } catch (Exception e) {
            e.printStackTrace();
            resp = e.getMessage();
        }
        return resp;
    }
    @Override
    protected void onPostExecute(String result) {
        progressDialog.dismiss();
        finalResult.setText(result);
    }
    @Override
    protected void onPreExecute() {
        progressDialog = ProgressDialog.show(MainActivity.this,
            "ProgressDialog", "Wait for "
+time.getText().toString()+"seconds");
    }
}
```

```
@Override  
protected void onProgressUpdate (String text)  
{  
    finalResult.setText(text[0]);  
}  
}
```

using `getText()` and `toString()` method and match it with the text using `equals()` function.
 The last thing you need to do is to provide a security mechanism, so that unwanted attempts should be avoided. For this initialize a variable and on each false attempt, decrement it. And when it reaches to 0, disable the login button.

VIII. Resources required(Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain the use of `equals()` function.
2. List the important functions which are related to GUI component "Button".
3. State the uses of Toast message.

(Space for answers)

Q1)

The `equal()` function is used to compare two strings

Q2)

-
- 1) `setEnabled()`
 - 2) `setOnClickListener()`
 - 3) `isSelected()`
 - 4) `animate()`
 - 5) `append()`

Q3)

→ A Toast is feedback message it takes a very little space for displaying while overall activity is interactive & visible to user

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to create the login form and display login successful/ Unsuccessful toastmessage.

(Space for answers)

```
→ package com.shruti.Exp27;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity
{
    EditText unname, pswd;
    Button login;
    DbHandler db;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        unname=(EditText)findViewById(R.id.edittext);
        pswd=(EditText)findViewById(R.id.edittext);
        login.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {

```



```
<EditText
```

```
android:id="@+id/e1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:ems="10"/>
```

```
<TextView
```

```
android:id="@+id/1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Surname"/>
```

```
<EditText
```

```
android:id="@+id/e2"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:layout_ems="10"/>
```

```
<EditText
```

```
android:id="@+id/e3"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:ems="10"/>
```

```
<Button
```

```
android:id="@+id/b1"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="ADD DATA"
```

```
<Button
```

```
android:id="@+id/b2"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="VIEW"/>  
</LinearLayout>
```

- Clean user interface.
- Validation (check if the email is an email and if the user entered all the data).
- Notifications for the user that the data is incorrect.
- Instructions for the user (e.g. how many characters are required for password).

VIII. Resources required (Additional)

Sr. No.	Instrument / Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain validation of user input?
2. List and explain various GUI components used to design the login form with validation.
3. Differentiate between Text View and Edit Text View.

(Space for answers)

1) → Input validation eliminates the errors that can be done by user while giving inputs to our app. For example if we want to get the user's email we can check the entered email is a valid email or not before storing it inside the database.

- 2) →
- 1) A field to fill in an email address.
 - 2) A field to fill in a password.
 - 3) A password visibility toggle that toggles the password visibility.
 - 4) Furthermore we want our field to have an error state that is activated when the input doesn't conform with email & password validation respectively.

3) →

EditText is used for user input. TextView is used to display text and is not editable by the user. TextView is the widget used when you want the user to view the text. and EditText used when you want you want the user to be able to edit the text.



X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)
1. Write a program to create the login form with necessary validations like length of username and password, empty text fields, count of unsuccessful login attempts. Display the login successful/Unsuccessful toastmessage.

(Space for answers)

```

package com.example.finalproject;
import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class LoginActivity extends Activity
implements OnClickListener{
    Button mlogin;
    Button mNewUser;
    Button mSignUp;
    EditText mUsername;
    EditText mPassword;
    SQLiteDatabase mdbh = null;
    private String uname;
    @Override
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        mNewUser = (Button) findViewById(R.id.buttonNewUser);
        mlogin = (Button) findViewById(R.id.buttonLogin);
        mlogin.setOnClickListener(this);

```

```

mShowAll = (Button)findViewById(R.id.ButtonShowAll)
mShowAll.setOnClickListener(this);
}
public void onClick(View v) {
    SQLite Database db = new SQLiteAdapter(Login
    Activity.this).openWrite();
    if (v.getId() == R.id.buttonlogin) {
        mUsername = (EditText)findViewById(R.id.password);
        mPassword = (EditText)findViewById(R.id.password);
        String username = mUsername.getText().toString();
        String pass = mPassword.getText().toString();
        if (username.equals("") || username == null) {
            Toast.makeText(getApplicationContext(), "
            Username Empty" Toast.LENGTH_SHORT).show();
        }
        else if (pass.equals("") || pass == null) {
            Toast.makeText(getApplicationContext(), "Password Empty"
            Toast.LENGTH_SHORT).show();
        }
        Cursor c = db.rawQuery ("SELECT email FROM MY -
        USERS TABLE WHERE email=? AND password=?",
        new String[] {username, pass});
        if (c.moveToFirst()) {
            Toast.makeText(getApplicationContext(), "Success",
            Toast.LENGTH_SHORT).show();
            Intent main = new Intent (LoginActivity.this,
            MainActivity.class);
            startActivity(main);
        }
        else {
            Toast.makeText(getApplicationContext(), "Failed
            to login Again", Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

To receive SMS messages, the best practice is to use the `onReceive()` method of the Broadcast Receiver class. The Android framework sends out system broadcasts of events such as receiving an SMS message, containing intents that are meant to be received using a Broadcast Receiver. Your app receives SMS messages by listening for the `SMS_RECEIVED_ACTION` broadcast.

Methods :

- 1 `ArrayList<String> divideMessage(String text) :-` This method divides a message text into several fragments, none bigger than the maximum SMS message size.
- 2 `static SmsManager getDefault() :-` This method is used to get the default instance of the Sms Manager
- 3 `void sendDataMessage(String destination Address, String scAddress, short destinationPort, byte[] data, PendingIntent sentIntent, PendingIntent deliveryIntent) :-` This method is used to send a data based SMS to a specific application port.
- 4 `void sendTextMessage(String destinationAddress, String scAddress, String text, PendingIntent sentIntent, PendingIntent deliveryIntent) :-` Send a text based SMS.

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain the use of `SmsManagerClass`.
2. List changes that are need to be done in `AndroidManifest.XML` file to send and receive messages.

(Space for answers)

Q1) → The `SmsManagerClass` manager sms operations such as sending data, text & pdu sms messages. we can get its object by calling static method `getDefault()`

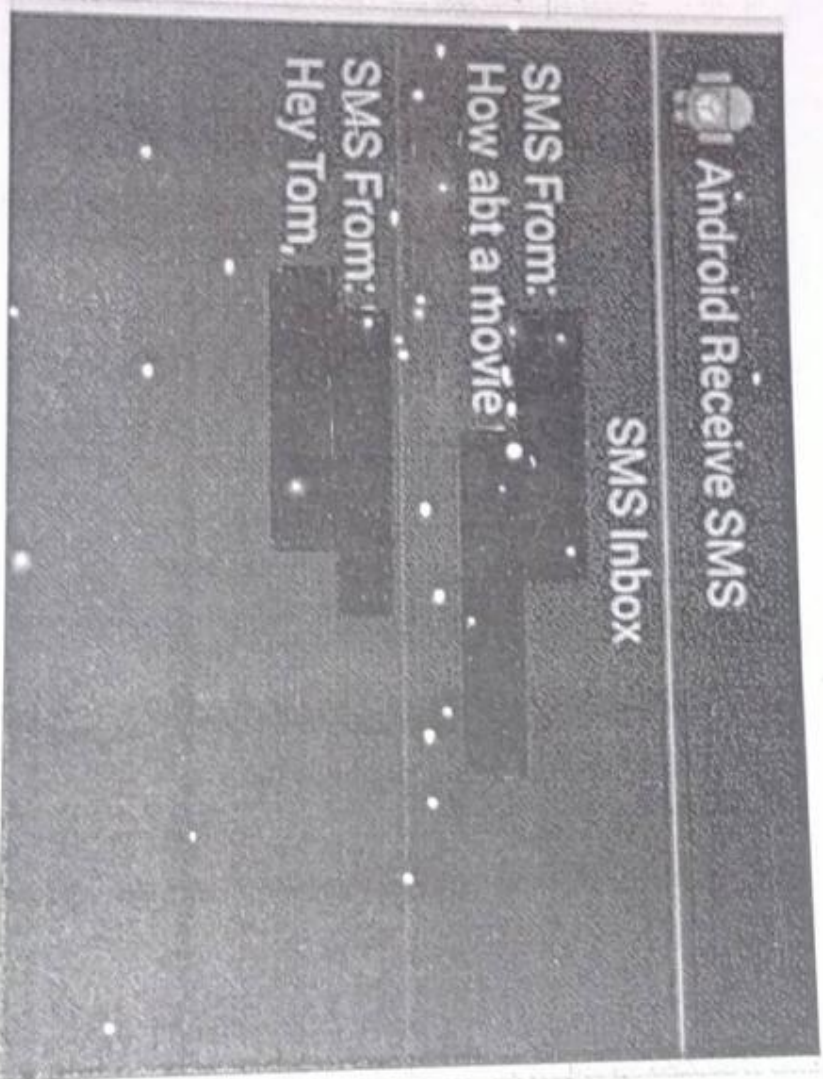
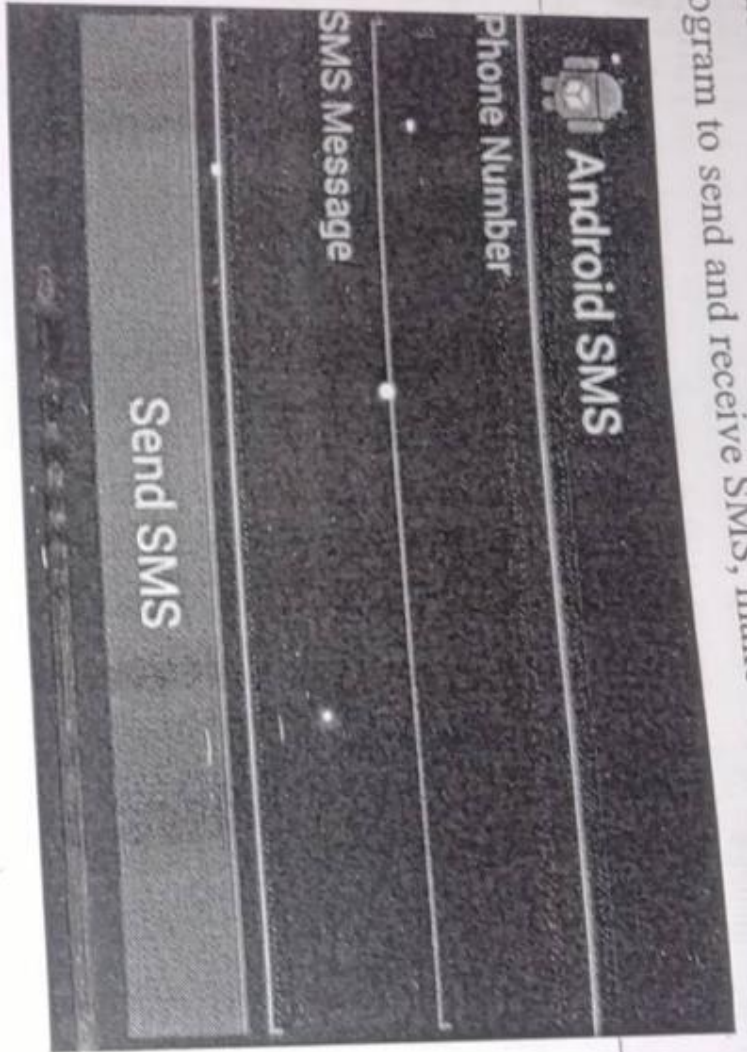
Q2)

→) <uses-permission android:name="android.permission.SEND_SMS"/>

2) <uses-permission android:name="android.permission.RECEIVE_SMS"/>

X. Exercise
Faculty must ensure that every group of students use different input value.

- Note: Faculty must ensure that every group of students use different input value.
(Use blank space for answers or attach more pages if needed)
1. Write a program to send and receive SMS, make use of following GUI.



(Space for answers)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:schemas="android:apk:tools"
    android:layout="width="match-parent"
    android:layout="height="match-parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textview1"
        public class MainActivity extends AppCompatActivity >
```

```

}
EditText e1, e2;
Button b1;
@Override
protected void onCreate (Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);
    setContentView (R.layout.activity_main);
    e1 = (EditText) findViewById (R.id.editText1);
    e2 = (EditText) findViewById (R.id.editText2);
    b1 = (Button) findViewById (R.id.button1);
    b1.setOnClickListener (new View.OnClickListener() {
        @Override
        public void onClick (View v) {
            String num = e1.getText ().toString ();
            String msg = e2.getText ().toString ();
            Intent i = new Intent (getApplication ().
                Context () MainActivity.class);
            PendingIntent pi = PendingIntent.
                getActivity (getApplication (). Context (). MainActivity.
                class)
            PendingIntent pi = PendingIntent.
                getActivity (getApplication (). Context (). MainActivity.
                class)
            SmsManager sms = SmsManager.getDefault ();
            android:layout_width = "match-parent"
            android:layout_height = "match-parent"
            android:text = "Number" />
        <EditText
            android:id="@+id/editText1"
            android:layout_width="wrap-content"
            android:layout_height="wrap-content"
            android:text="Number"
            android:ems="10" />
        <TextView
            android:id="@+id/textView2"
    
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Message" />
<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="SEND" />
</LinearLayout>
```

VIII. Resources required (Additional)

Sr. No.	Instrument / Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Why it becomes necessary to have inbuilt email module in mobile applications.
2. List the extra fields that can be used in an application to send emails.

(Space for answers)

1) If we have an inbuilt email module it becomes easier & faster to send mails rather than just browsing to email website then logging in into it & then sending mail.

- 2)
- 1) ACTION_SENDTO
 - 2) EXTRA_BCC
 - 3) EXTRA_CC
 - 4) EXTRA_EMAIL
 - 5) EXTRA_HTML_TEXT
 - 6) EXTRA_SUBJECT
 - 7) EXTRA_TEXT
 - 8) EXTRA_TITLE

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to send email.

(Space for answers)

```

-> Intent email = new Intent(Intent.ACTION_SEND);
    email.putExtra(Intent.EXTRA_EMAIL, new String[]
    { "rs" });
    email.putExtra(Intent.EXTRA_SUBJECT, subject);
    email.putExtra(Intent.EXTRA_TEXT, message);
    email.setType("message/rfc822");
    startActivity(Intent.createChooser(email,
    "Choose an Email client:"));

MainActivity.java
package com.example.sendmail;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.os.Bundle;
public class MainActivity extends Activity {
    EditText editText;
    EditText subject;
    EditText message;
    Button send;

    @Override
    public class MainActivity extends Activity {
        EditText editText;
        EditText subject;
        EditText message;
        Button send;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_main);
editTextTo = (EditText) findViewById(R.id.editText);
editTextSubject = (EditText) findViewById(R.id.editTextSubject);
editTextMessage = (EditText) findViewById(R.id.editTextMessage);
send = (Button) findViewById(R.id.button);
send.setOnClickListener(new OnClickListener() {
    @Override
    
```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	20%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	

```
public void onClick(View arg0) {  
    String to = editTextTo.getText().toString();  
    String subject = editTextSubject.getText().toString();  
    String message = editTextMessage.getText().toString();  
    Intent email = new Intent(Intent.ACTION_SEND);  
    email.putExtra(Intent.EXTRA_EMAIL, new String[]{"to"});  
    email.putExtra(Intent.EXTRA_TEXT, message);  
    email.setType("message/rfc822");  
    startActivity(Intent.createChooser(email, "Choose an  
    Email client:"));  
}  
}  
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.activity_main, menu);  
    return true;  
}  
}
```



```
<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_STO
RAGE" />
```

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List the names of map type and write the syntax to change it.
2. Name the methods used to enable and disable zoom feature.

(Space for answers)

- 1) Normal
2) Hybrid
3) Satellite
4) Terrain
5) None

Syntax - GoogleMap map

map.setType(GoogleMap.MAP_TYPE);

2) → m.Map.addMarkers (NewMarketOptions) position (Sydney), title ("Market in Sydney");
m.Map.moveCamera (camera (camera Update Factory.newInstance (Sydney)));

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to locate user's current location.

(Space for answers)

→ package example.com.mapexample;

import android.os.Bundle;

import android.support.v4.app.FragmentActivity;

import android.os.Bundle;

public class MainActivity extends

FragmentActivity implements OnMap

ReadyCallback, LocationListener, GoogleApiClient

ConnectionCallbacks,

GoogleApiClient, OnConnectionFailedListener {

private GoogleMap mMap;

Location, LocationRequest;

Marker mCurrLocationMarker;

GoogleApiClient mGoogleApiClient;

LocationRequest mLocationRequest;

@Override

protected void onCreate (Bundle savedInstanceState)

{ super.onCreate (savedInstanceState);

setContentView (R.layout.activity_maps);

SupportMapFragment mapFragment =

(SupportMapFragment) getSupportFragmentManager

.findFragmentById (R.id.map);

mapFragment.getMapAsync (this)

@Override

public void onMapReady (GoogleMap googleMap)

{ mMap = googleMap;

f. android.os.Build.VERSION.SDK_INT >=

Build.VERSION_CODES.M {

f. ContextCompat.checkSelfPermission (this,

Manifest.permission.ACCESS_FINE_LOCATION

PackageManager.PERMISSION_GRANTED) {

...

...

...

```
buildGoogleApiClient();  
mMap.setMyLocationEnabled(true);  
}  
else {  
    buildGoogleApiClient();  
    mMap.setMyLocationEnabled(true);  
}  
protected synchronized void buildGoogleApiClient() {  
    mGoogleApiClient = new GoogleApiClient.Builder(this)  
        .addConnectionCallbacks(this)  
        .addOnConnectionFailedListener(this)  
        .addApi(LocationServices.API).build();  
    mGoogleApiClient.connect();  
}
```

```
@Override  
public void onConnected (Bundle bundle) {  
    mLocationRequest = new LocationRequest();  
    mLocationRequest.setFastestInterval(1000);  
    mLocationRequest.setInterval(1000);  
    mLocationRequest.setPriority (LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);  
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {  
        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);  
    }  
}
```

```
@Override  
public void onConnectionSuspended (int i) {  
}
```

```
@Override
public void onLocationChanged (Location location) {
    mLastLocation = location;
    if (mCurrentLocationMarker != null) {
        mCurrentLocationMarker.remove();
    }
    LatLng latLng = new LatLng (location.getLatitude(),
    location.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions ();
    markerOptions.title ("Current Position");
    markerOptions.icon (BitmapDescriptorFactory.
    defaultMarker (BitmapDescriptorFactory.HUE_GREEN));
    mMap.setLocationMarker (mMap.addMarker (markerOptions));
    mMap.moveCamera (CameraUpdateFactory.new
    LatLng (latLng));
    mMap.animateCamera (CameraUpdateFactory
    .zoomTo (11));
    if (mGoogleApiClient != null) {
        LocationServices.FusedLocationApi.remove
        LocationUpdates (mGoogleApiClient, this);
    }
}

@Override
public void onConnectionFailed (Connection
Result connectionResult) {
}
```

VIII. Resources required (Additional)

Sr. No.	Instrument /Object	Specification	Quantity	Remarks
1	Android enabled smartphone / Android version supporting emulator	2 GB RAM	1	Data cable is mandatory for emulators

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Explain the ways to add Markers on the Google Map.
2. Write the syntax for method which is used to add compass in Google Map.

(Space for answers)

1) i) Adding a pin on mobile

ii) Adding to your places on mobile

iii) Using the Desktop website

iv) Creating a map with MyMaps on Android

v) Creating a map with MyMaps on Desktop

vi) Viewing MyMaps on Google Maps

2) i) Setting set Compass Enabled (Choolan)

X. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a program to draw a route between two locations.

(Space for answers)

```

package com.wingsquare.routebetweenlocation;
import com.google.android.gms.maps.
mode.LatLng;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
public class DirectionsJSONParser
public List<List<HashMap<String,String>>>parse
(JSONObject object) {
List<List<HashMap<String,String>>> routes =
new ArrayList<>();
JSONArray jRoutes = null;
JSONArray jLegs = null;
JSONArray jSteps = null;
try {
jRoutes = object.getJSONArray("routes");
for (int i=0; i<jRoutes.length(); i++) {
jLegs = (JSONObject)jRoutes.get(i).getJSONArray
("legs");
List path = new ArrayList<HashMap<String,String>>();
for (int j=0; j<jLegs.length(); j++) {
jSteps = (JSONObject)jLegs.get(j);
for (int k=0; k<jSteps.length(); k++) {
String polyline = "";
polyline = (String) ((JSONObject)jStep
.getJSONObject("polyline")).get("points");
List<LatLng> list = decodePoly(polyline);
for (int l=0; l<list.size(); l++) {

```

```

HashMap<String, String> hm = new HashMap<
String, String>(); Double toString((latLng)
hm.put("lat", latitude);
list.get(0).latitude);
hm.put("lng", Double.toString((latLng).list
.get(0).longitude)); Reading

```

XI. References / Suggestions for further Reading

1. <https://www.tutorialspoint.com/android>
2. <https://stuff.mit.edu>
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators		Weightage
Process related (10 Marks)		30%
1.	Logic Formation	10%
2.	Debugging ability	15%
3.	Follow ethical practices	5%
Product related (15 Marks)		70%
4.	Interactive GUI	20%
5.	Answer to Practical related questions	20%
6.	Expected Output	20%
7.	Timely Submission	10%
Total (25 Marks)		100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related (10)	Product Related (15)	Total (25)	

```

HashMap<String,String> hm = new HashMap<
String,String>();
String s1="lat", s2="double", s3="lon";
hm.put(s1,s2);
hm.put(s1,"lat");
hm.put(s1,"lon");
hm.put(s1,"lon");
hm.put(s1,"lon");
get(s1).longitude); Reading
References / Suggestions for further
1. https://www.tutorialspoint.com/android
2. https://stuff.mit.edu
3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
4. https://developer.android.com

```

- XI. References / Suggestions for further Reading
1. <https://www.tutorialspoint.com/android>
 2. <https://stuff.mit.edu>
 3. https://www.tutorialspoint.com/android/android_advanced_tutorial.pdf
 4. <https://developer.android.com>

XII. Assessment Scheme

Performance indicators	Weightage
Process related (10 Marks)	30%
1. Logic Formation	10%
2. Debugging ability	15%
3. Follow ethical practices	5%
Product related (15 Marks)	70%
4. Interactive GUI	20%
5. Answer to Practical related questions	20%
6. Expected Output	20%
7. Timely Submission	10%
Total (25 Marks)	100%

List of student Team Members

1.
2.
3.
4.

Marks Obtained			Dated signature of Teacher
Process Related(10)	Product Related(15)	Total (25)	


```
path.add(Chm);??  
routes.add(path);  
??}  
catch (JSONException e) {  
    e.printStackTrace();  
} catch (Exception e) {  
}  
return routes;  
}  
  
private List<LatLng> decodePoly (String encoded) {  
    List<LatLng> poly = new ArrayList<LatLng>();  
    int index = 0, len = encoded.length();  
    int lat = 0, lng = 0;  
  
    while (index < len) {  
        int b, shift = 0, result = 0;  
        do {  
            b = encoded.charAt(index++) - 63;  
            result |= (b <& 0x0f) << shift;  
            shift += 5;  
        } while (b >= 0x20);  
        int dlat = (result <& 0xffff) << 1;  
        if (result >> 1) {  
            shift = 0;  
            result = 0;  
        }  
        do {  
            b = encoded.charAt(index++) - 63;  
            result |= (b <& 0x0f) << shift;  
            shift += 5;  
        } while (b >= 0x20);  
        int dlat = (result <& 0xffff) << 1;  
        if (result >> 1) {  
            shift = 0;  
            result = 0;  
        }  
        b = encoded.charAt(index++) - 63;  
        result |= (b <& 0x0f) << shift;  
        shift += 5;  
    }  
}
```

```
{ while(b >= 0x20);  
int dlng = ((result &amp;amp;P, 1) != 0 ?  
!(result >> 1) && result >> 1));  
lng += dlng;  
latlngp = new Latlng((double)lat / 1E5));  
C((double)lng / 1E5));  
poly.add(P);  
return poly;  
}
```