



Maharashtra State Board of Technical Education

Certificate

This is to certify that Mr. / Ms. Siddharth Mahendra Revankar

Roll No. 35 of Sixth Semester of Diploma in
Computer Engineering of Institute
VPM's Polytechnic, Thane (Code 0007....)

has completed the term work satisfactorily in subject **Mobile Application Development (22617)** for the academic year 20.20...to 20.21.... as prescribed in the curriculum.

Place ..Thane

Enrollment No. 1800070101.....

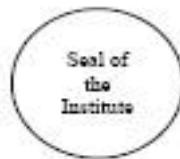
Date: 10/06/2021

Exam Seat No. 103787.....

Subject Teacher

Head of the Department

Principal



Content Page

List of Practical's and Progressive Assessment Sheet.

S. No.	Title of the practical	Page No.	Date of performance	Date of submission	Assessment marks (25)	Dated sign.of teacher	Remarks(if any)
1.	Compare various operating systems with Android OS.	1	05/04/21	09/04/21	22		
2.	Install and configure java development kit(JDK), android studio and android SDK.	5	05/04/21	09/04/21	24		
3.	Configure android development tools (ADT)plug-in and create android virtual device.	9	05/04/21	09/04/21	24		
4.	Develop a program to display Hello World on screen.	14	14/04/21	15/04/21	24		
5.	Develop a program to implement linear layout and absolute layout.	20	14/04/21	19/04/21	24		
6.	Develop a program to implement frame layout, table layout and relative layout.	26	14/04/21	28/04/21	25		
7.	Develop a program to implement Text View and Edit Text.	31	15/04/21	29/04/21	24		
8.	Develop a program to implement AutoComplete Text View.	36	15/04/21	29/04/21	24		
9.	Develop a program to implement Button, Image Button and Toggle Button.	41	15/04/21	29/04/21	25		
10.	Develop a program to implement login window using above UI controls.	48	21/04/21	02/05/21	24		
11.	Develop a program to implement Checkbox.	54	21/04/21	02/05/21	24		
12.	Develop a program to implement Radio Button and Radio Group.	59	22/04/21	02/05/21	24		
13.	Develop a program to implement Progress Bar.	64	22/04/21	02/05/21	24		
14.	Develop a program to implement List View, Grid View, Image View and Scroll View.	69	22/04/21	05/05/21	24		
15.	Develop a program to implement Custom Toast Alert.	76	28/04/21	05/05/21	24		
16.	Develop a program to implement Date and Time Picker.	82	28/04/21	05/05/21	24		

17.	Develop a program to create an activity.	88	28/04/21	05/05/21	24		
18.	Develop a program to implement new activity using explicit intent and implicit intent.	94	29/04/21	10/05/21	24		
19.	Develop a program to implement content provider.	100	29/04/21	10/05/21	24		
20.	Develop a program to implement service.	105	29/04/21	10/05/21	24		
21.	Develop a program to implement broadcast receiver.	110	05/05/21	10/05/21	24		
22.	Develop a program to implement sensors.	115	05/05/21	17/05/21	24		
23.	Develop a program to build Camera.	121	06/05/21	17/05/21	24		
24.	Develop a program for providing Bluetooth connectivity.	126	06/05/21	17/05/21	24		
25.	Develop a program for animation.	131	13/05/21	24/05/21	24		
26.	Perform Async task using SQLite.	136	19/05/21	24/05/21	25		
27.	Create sample application with login module. (Check username and password) On successful login, Change Text View “Login Successful” and on login fail, alert user using Toast “Login fail”.	141	26/05/21	02/06/21	25		
28.	Create login application where you will have to validate username and password till the username and password is not validated, login button should remain disabled.	146	02/06/21	02/06/21	25		
29.	Develop a program to a) Send SMS b) Receive SMS	151	02/06/21	02/06/21	25		
30.	Develop a program to send and receive e-mail.	157	03/06/21	09/06/21	25		
31.	Deploy map-based application. Part I	161	03/06/21	09/06/21	25		
32.	Deploy map-based application. Part II	165	09/06/21	09/06/21	25		
Total Marks					775		
Total Marks (Scaled to 25 Marks)					24		

Practical 1

Aim: Compare various operating systems with Android OS.

* Practical Related Questions.

i) List different Android O.S. versions.

Version	Codenam	APK
1.0		1
1.1		2
1.5	Cupcake	3
1.6	Donut	4
2.0	Eclair	5
2.0.1	Eclair	6
2.1	Eclair	7
2.2x	Eriyo	8
2.2 - 2.3.2	Gingerbread	9
2.3.2 - 2.3.7	Gingerbread	10
3.0	Honeycomb	11
3.1	Honeycomb	12
3.2	Honeycomb	13
4.0 - 4.0.2	Ice cream Sandwich	14
4.0.3 - 4.0.4	Ice cream Sandwich	15
4.1	Jelly Bean	16
4.2	Jelly Bean	17
4.3	Jelly Bean	18
4.4	KitKat	19
4.4	KitKat	20
5.0	Lollipop	21

2] Static characteristics of Android Operating System.

- i) Beautiful UI
- 2) Connectivity
- 3) Storage
- 4) Media Support
- 5) Messaging
- 6) Web Browser
- 7) Multi-Touch

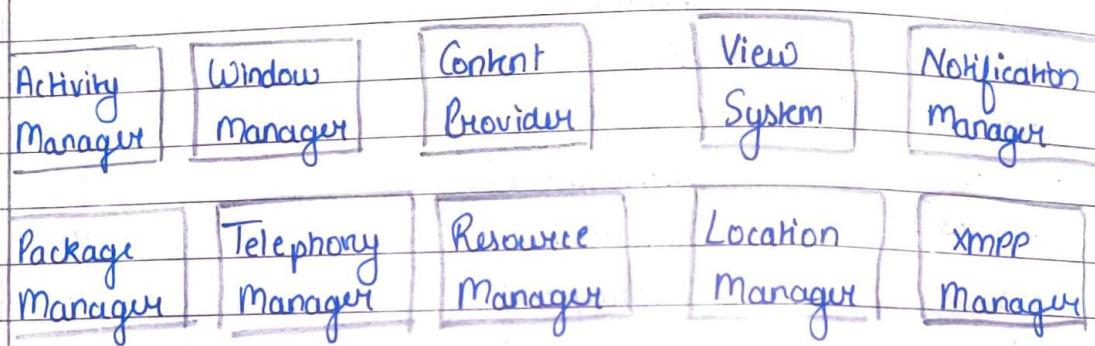
- 8) Multi-tasking
- 9) Reusable Widgets
- 10) Multilanguage
- 11) Gcm
- 12) Wi-fi direct
- 13) Android Beam

* Exercise

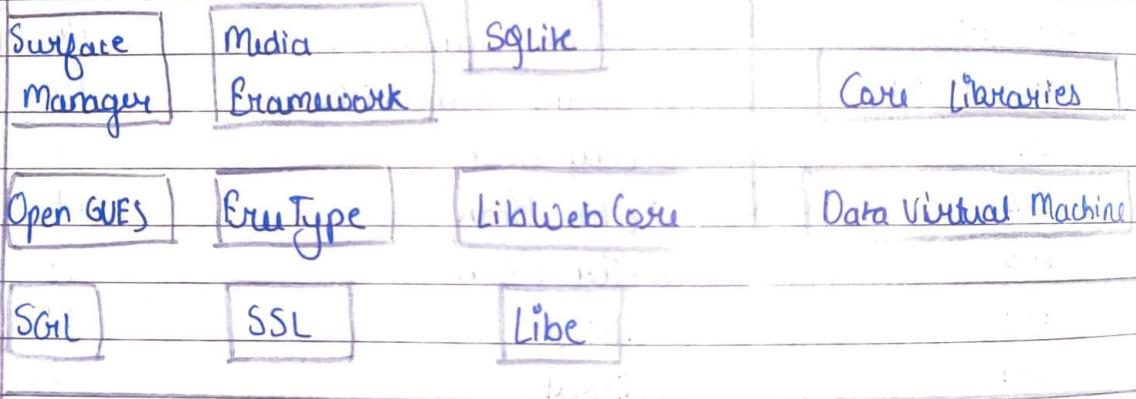
i) Draw the architectural diagram of Android Operating System.



Application Framework



Libraries



Android Runtime

Core Libraries



2) Differentiate between Windows operating system & Android Operating system.

Parameter	Android	Windows
Company	Google	Microsoft
Programmed in	Java, C, C++	C, C++
OS Family	Unix - like	Windows
Initial Release	September 23, 2008	October 21, 2010
Supported Platforms	ARM, mips, x86, I-mx	x86
License	Apache License 2.0 Linux kernel patches under GNU GPL v2	Commercial proprietary software
UI	Graphical (multi-touch)	Graphical (metro - UI)

Practical 2

Aim: Install & configure Java development kit (JDK),
Android Studio & android SDK.

* Practical Related Questions.

- ij List all the steps to install android operating system.
→ ij Step 1: Setup Java Development Kit (JDK) by downloading & installing latest version of Java JDK from Oracle's Java site: Java SE Downloads.
- 2j Step 2: Setup Android SDK by downloading the latest version of android SDK from android's official website <http://developer.android.com/sdk/index.html>.

ij List all the steps to install android operating system.

→ Step 1: Download Google Android OS.

Step 2: Write ISO Image file to CD/DVD or USB drive.

Step 3: Boot your computer using CD/DVD or USB

Step 4: Install or test run Google Android in your computer.

Step 5: Running Google Android OS in your PC.

2] List various IDE's that can be used to execute android operating system.

→ 1] ADT

2] DroidScript

3] CppDroid

4] Android Web Development

5] Python Suite

6] Java Suite

7] Eclipse

8] Android Studio

9] Arduino

10] Visual Studio.

* Exercise

→ 1] Differentiate between JVM & DVM.

JVM

1] Stack based

2] Uses "java" byte code & runs ".class" file having JIT

3] Single instance of JVM is shared with multiple applications

4] Supports multiple OSs.

5] Many Re-tools are available

6] Constant pool for every class.

7] Executable is JAR

DVM

1] Register based which is designed to run on low memory.

2] Uses its own byte code & runs ".dex" file.

3] Has been designed so that a device can run multiple instances of VM efficiently.

4] Supports only Android OS.

5] Very few Re-tools are available.

6] Constant pool for every application.

7] Executable is APK.

2] What is IDE? Why Java development kit is essential to install an android operating system?

→ An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. JDK is essential because android studio supports Kotlin, & Kotlin is for JDK compilation & also android uses JDK to compile android java to virtual machine language.

Practical 3:

Aim: Configure android development tools (ADT) plug-in & create android virtual device.

* Practical Related Questions:

i) List basic requirements for configuring android operating system.

→ 1] Microsoft Windows 7/8/10 (32/64 bit)

2] 3 GB RAM minimum, 8GB RAM recommended.

3] 2 GB available disk space minimum, 9GB recommended.

4] 1280 x 800 minimum screen resolution.

2] Why bytecode cannot run in android?

→ The class file i.e. the byte code for android is first optimized even more to make it mobile friendly, which differs from normal bytecode. Hence, direct Java bytecode won't run.

* Exercise:

i) What is a Build Type in Gradle?

→ Android studio uses Gradle; an advanced build toolkit, to automate & manage the build process while allowing you to define flexible custom build configuration. Each build configuration can define its own set of code & resources while reusing the parts common to all version of your app.

2] Explain the build process in Android.

→ 1] The compiler converts your source code into DEX files, which include the bytecode that runs on Android devices & everything else into compiled resources.

2] The APK package combines the DEX files & compiled resources into a single APK. Before your app can be installed & developed onto an Android device.

3] However, the apk package signs your APK using either debug or release. 4] Before generating your final APK, the packager signs your zipalign tool to optimize your app.

Practical 4

Develop a program to display Hello World on screen.

Siddharth Revankar TYCO

I. Practical Significance

In android studio students must be aware of the directory structure and the control flow the program. Program should be either executed on the android mobile phones or on the suitable emulators. To execute a simple program, like to display Hello World on screen syntax of writing a program in android is pre-requisite as the programming language used is JAVA only. The main activity code is a Java file MainActivity.java. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application.

II. Minimum Theoretical Background

The Manifest File: Whatever component you develop as a part of your application, you must declare all its components in a manifest file called AndroidManifest.xml which resides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS. The <activity> tag is used to specify an activity and android:name attribute specifies the fully qualified class name of the Activity subclass and the android:label attribute specifies a string to use as the label for the activity. You can specify multiple activities using <activity> tags. The action for the intent filter is named android.intent.action.MAIN to indicate that this activity serves as the entry point for the application. The category for the intentfilter is named android.intent.category.LAUNCHER to indicate that the application can be launched from the device's launcher icon. The @string refers to the strings.xml file explained below. Hence, @string/app_name refers to the app_name string defined in the strings.xml file, which is "HelloWorld". Similar way, other strings get populated in the application. Following is the list of tags which you will use in your manifest file to specify different Android application components:

- 1) <activity> elements for activities
- 2) <service> elements for services
- 3) <receiver> elements for broadcast receivers
- 4) <provider> elements for content providers

III. Practical Related Questions

1. List the files used to write Hello World program.

Ans. Files used are:

Activity_main.xml

MainActivity.java

2. What is an activity in Android programming?

Ans. The <activity> tag is used to specify an activity and android:name attribute specifies the fully qualified class name of the Activity subclass and the android:label attribute specifies a string to use as the label for the activity. You can specify multiple activities using <activity> tags

IV Exercise

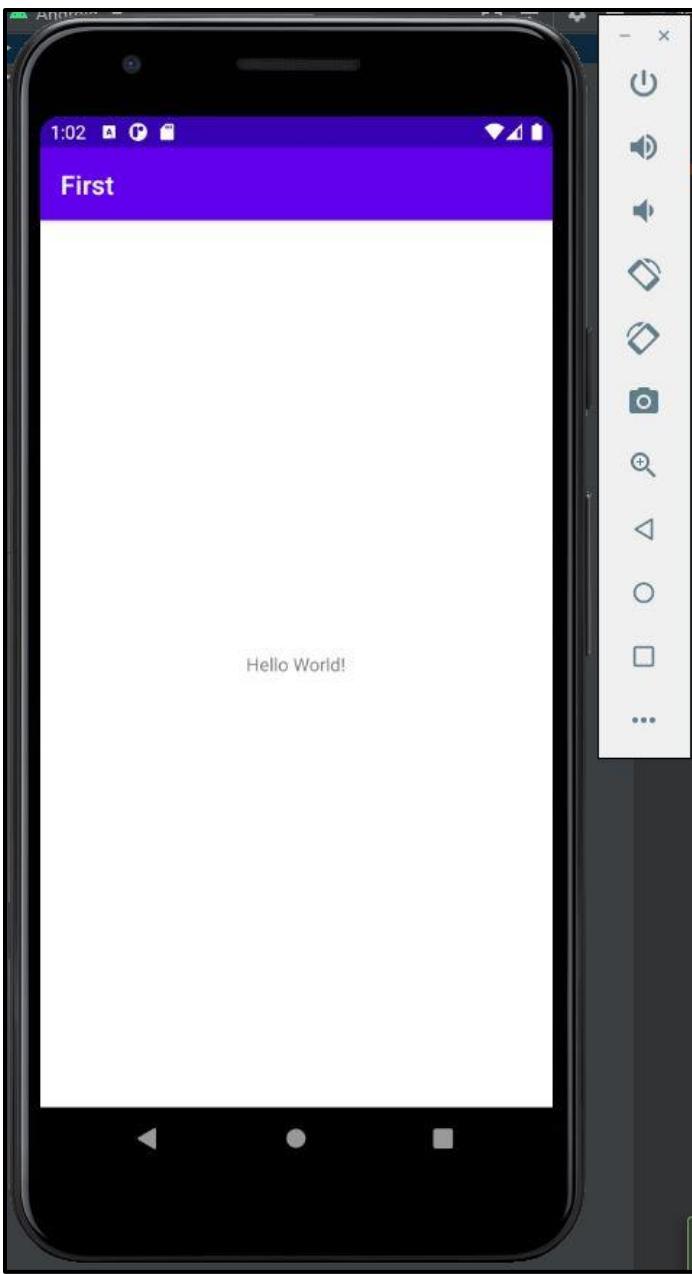
1. Write a program to display HelloWorld

Ans.

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



2. Write a program to display student name and marks

Ans.

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

```
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student Detalis"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.487"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.264" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="205dp"
        android:layout_height="35dp"
        android:layout_marginStart="96dp"
        android:layout_marginLeft="96dp"
        android:layout_marginTop="226dp"
        android:layout_marginEnd="111dp"
        android:layout_marginRight="111dp"
        android:layout_marginBottom="470dp"
        android:text="Siddharth Revankar"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="226dp"
        android:layout_marginEnd="331dp"
        android:layout_marginRight="331dp"
        android:layout_marginBottom="475dp"
        android:text="Name:"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
```

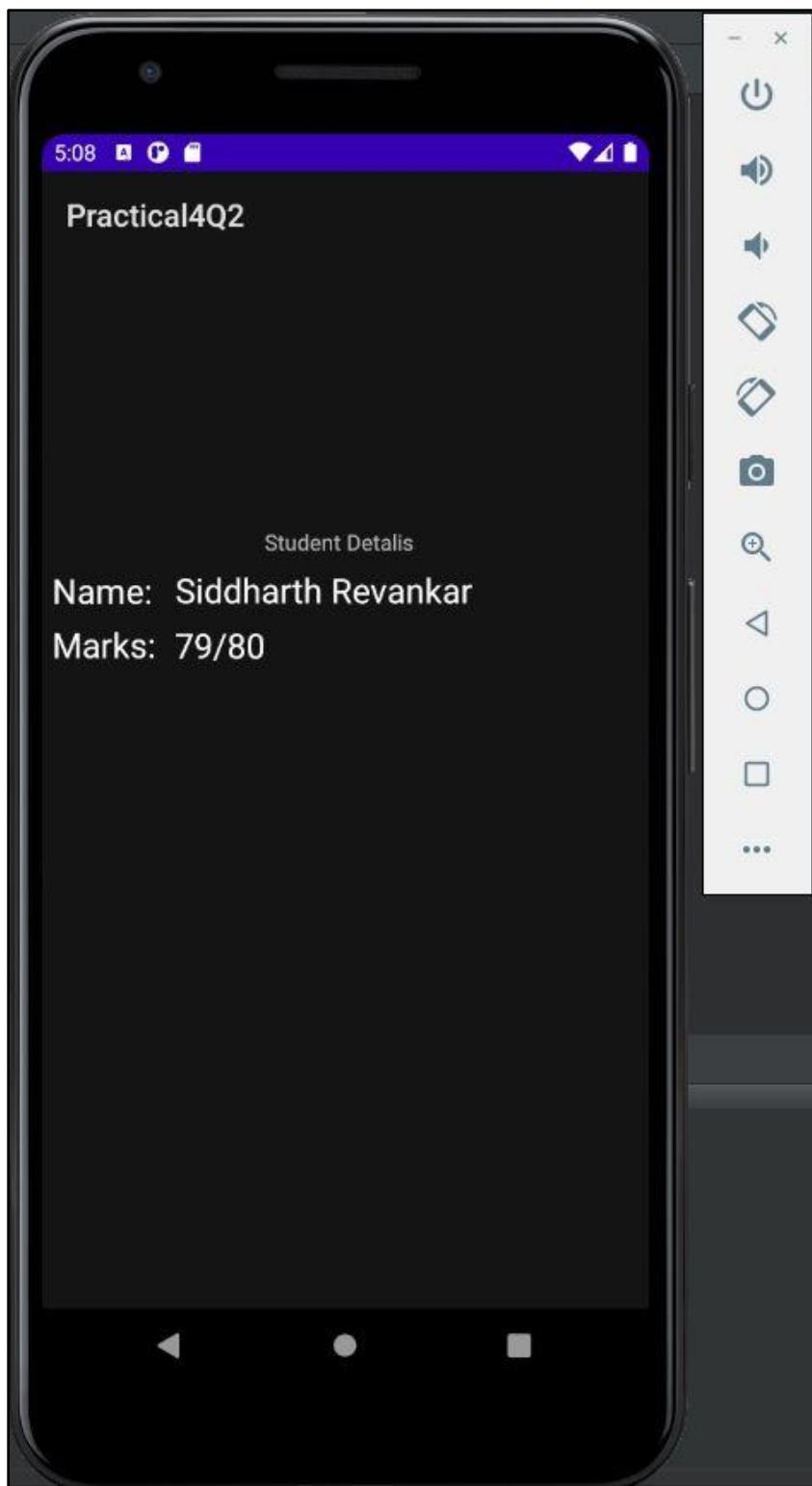
```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="261dp"
    android:layout_marginEnd="323dp"
    android:layout_marginRight="323dp"
    android:layout_marginBottom="440dp"
    android:text="Marks: "
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="95dp"
    android:layout_marginLeft="95dp"
    android:layout_marginTop="262dp"
    android:layout_marginEnd="257dp"
    android:layout_marginRight="257dp"
    android:layout_marginBottom="440dp"
    android:text="79/80"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

Output:



Practical 5

Develop a program to implement linear layout and absolute layout

Siddharth Revankar TYCO

I. Practical Significance

To develop and place the android components accurately on the display screen, android provides various layout managers. Layout managers can be used on the simple android program too. Various layout managers can be selected as per the program requirements

II. Practical Related Questions

1. **Name any three layout managers.**

Ans. Layout managers are:

- Linear Layout Manager
- Absolute Layout Manager
- Relative Layout Manager
- Constraint Layout Manager

2. **What is card view?**

Ans. CardView is a new widget in Android that can be used to display any sort of data by providing a rounded corner layout along with a specific elevation. CardView is the view that can display views on top of each other. The main usage of CardView is that it helps to give a rich feel and look to the UI design.

III. Exercise

1. **Write a program to place Name, Age and mobile number linearly (Vertical) on the display screen using Linear layout.**

Ans.

Activity_main.xml:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="#FFE500"  
    android:orientation="vertical">
```

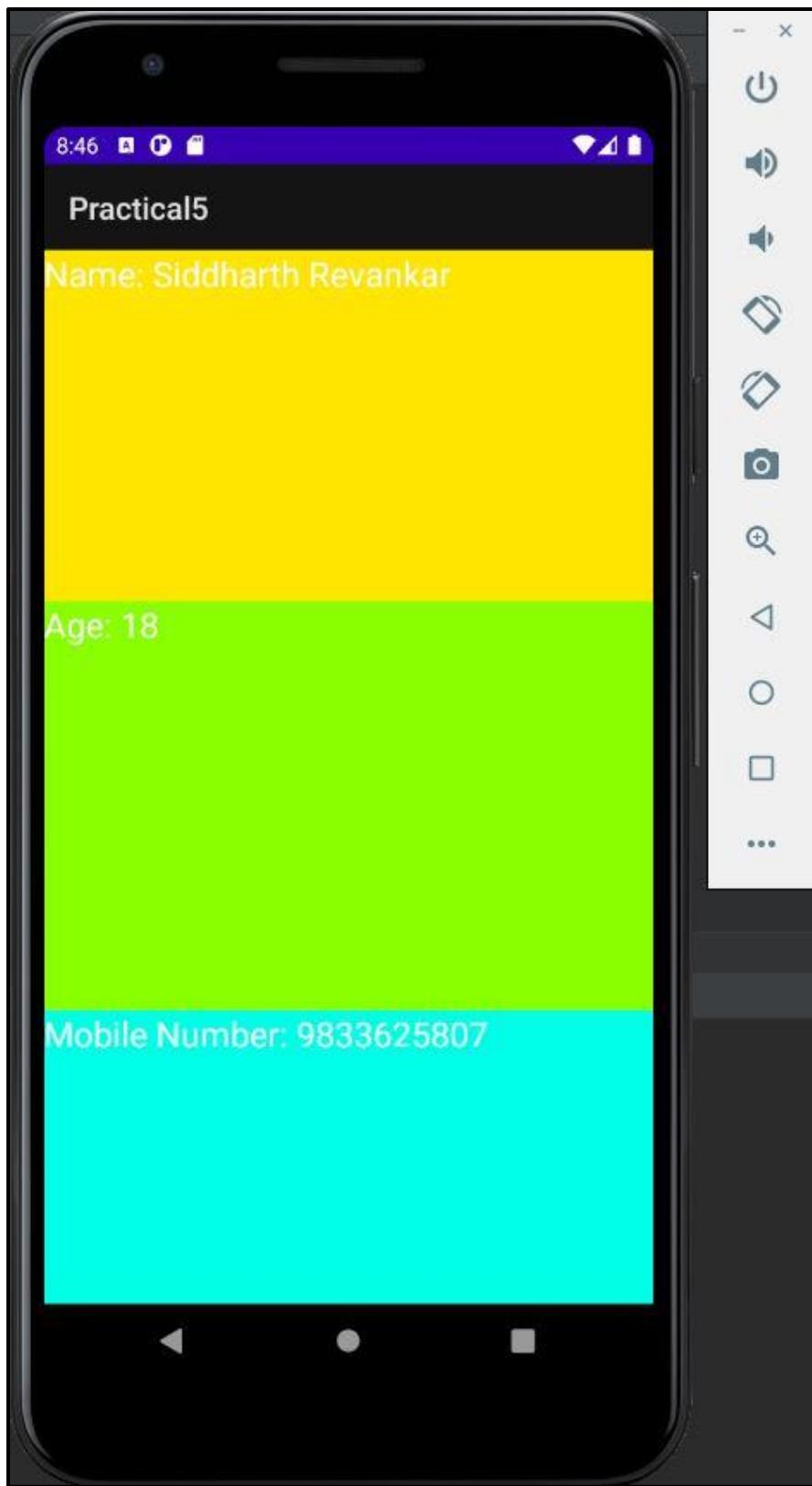
```
<TextView
    android:layout_width="match_parent"
    android:layout_height="226dp"
    android:text="Name: Siddharth Revankar"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="264dp"
    android:background="#89FF00"
    android:text="Age: 18"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="243dp"
    android:background="#00FFE7"
    android:text="Mobile Number: 9833625807"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

</LinearLayout>
```

Output:



2. Write a program to place Name, Age and mobile number centrally on the display screen using Absolute layout.

Ans.

Activity_main.xml:

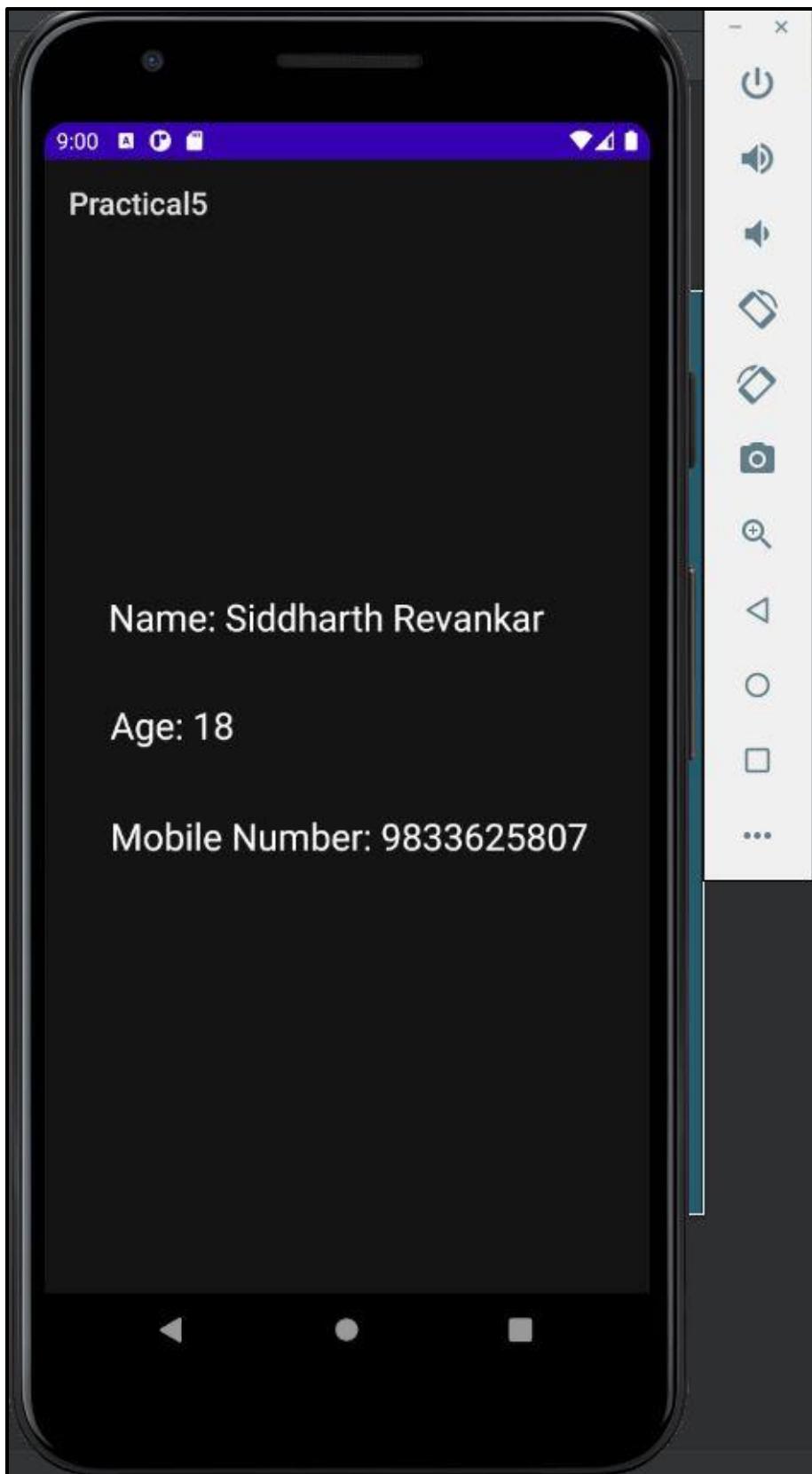
```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <TextView
        android:id="@+id/textView7"
        android:layout_width="322dp"
        android:layout_height="wrap_content"
        android:layout_x="42dp"
        android:layout_y="225dp"
        android:text="Name: Siddharth Revankar"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/textView8"
        android:layout_width="324dp"
        android:layout_height="wrap_content"
        android:layout_x="43dp"
        android:layout_y="295dp"
        android:text="Age: 18"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/textView9"
        android:layout_width="324dp"
        android:layout_height="wrap_content"
        android:layout_x="43dp"
        android:layout_y="367dp"
        android:text="Mobile Number: 9833625807"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textSize="24sp" />
</AbsoluteLayout>
```

Output:



Practical 6

Implement Develop a program to implement frame layout, table layout and relative layout.

Siddharth Revankar TYCO

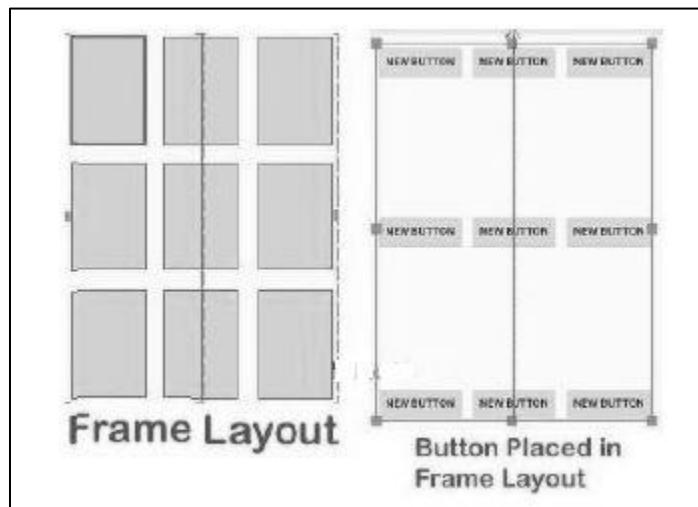
I. Practical Significance

To develop and place the android components accurately on the display screen, android provides various layout managers. Layout managers can be used on the simple android program too. Various layout managers can be selected as per the program requirements

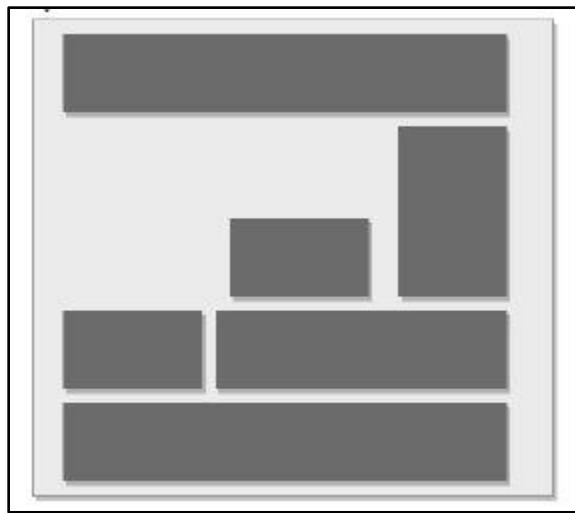
II. Minimum Theoretical Background

1. Frame Layout:

Frame Layout is designed to block out an area on the screen to display a single item. Generally, Frame Layout should be used to hold a single child view, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other. You can, however, add multiple children to a Frame Layout and control their position within the Frame Layout by assigning gravity to each child, using the android: layout gravity attribute. Child views are drawn in a stack, with the most recently added child on top. The size of the Frame Layout is the size of its largest child (plus padding), visible or not (if the Frame Layout's parent permits).



2. Relative Layout:



it can eliminate nested View groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested Linear Layout groups, you may be able to replace them with a single Relative Layout.

3. Table Layout:

A Table Layout consists of a number of Table Row objects, each defining a row (actually, you can have other children, Which Will be explained below). Table Layout containers do not display border lines for their rows, columns, or cells. Each row has zero or more cells; each cell can hold one View object. The table has as many columns as the row with the most cells. A table can leave cells empty. Cells can span columns, as they can in HTML the Width of a column is defined by the row With the Widest cell in that column.

Row 1		
Row 2 Column 1	Row 2 Column 2	Row 2 Column 3
Row 3		
Row 4 Column 1		Row 4 Column 2

III. Practical Related Questions

1. List different attributes which can be used with any layout manager.

Ans.

- **android:id:** It uniquely identifies the Android Layout.
- **android:hint:** It shows the hint of what to fill inside the EditText.
- **android:layout_height:** It sets the height of the layout.
- **android:layout_width:** It sets the width of the layout.
- **android:layout_gravity:** It sets the position of the child view.
- **android:layout_marginTop:** It sets the margin of the from the top of the layout.
- **android:layout_marginBottom:** It sets the margin of the from the bottom of the layout.
- **android:layout_marginLeft:** It sets the margin of the from the left of the layout.
- **android:layout_marginRight:** It sets the margin of the from the right of the layout.
- **android:layout_x:** It specifies the x coordinates of the layout.
- **android:layout_y:** It specifies the y coordinates of the layout.

2. What is Grid Layout?

Ans. A layout that places its children in a rectangular *grid*.

The grid is composed of a set of infinitely thin lines that separate the viewing area into *cells*. Throughout the API, grid lines are referenced by grid *indices*. A grid with N columns has N + 1 grid indices that run from 0 through N inclusive. Regardless of how GridLayout is configured, grid index 0 is fixed to the leading edge of the container and grid index N is fixed to its trailing edge (after padding is taken into account).

IV. Exercise

1. Write a program to display 10 students basic information in a table form using Table layout

Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
```

```
tools:context=".MainActivity">

<TableRow
    android:id="@+id/row1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#4927A5"
    android:visibility="visible"
    tools:visibility="visible">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="Roll No." />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="82dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="Name" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="79dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Percentage" />
</TableRow>

<TableRow
    android:id="@+id/row2"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView4"
        android:layout_width="52dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="1" />
```

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="121dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="Siddharth" />
```

```
<TextView
    android:id="@+id/textView6"
    android:layout_width="72dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:text="99" />
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="2" />
```

```
<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Prathamesh" />
```

```
<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="99" />
```

```
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">>

    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="3" />

    <TextView
        android:id="@+id/textView11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Tejashree" />

    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="99" />
</TableRow>

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

    <TextView
        android:id="@+id/textView13"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="4" />

    <TextView
        android:id="@+id/textView14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Vaibhav" />

    <TextView
        android:id="@+id/textView15"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="98" />
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <TextView
        android:id="@+id/textView16"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="5" />
```

```
    <TextView
        android:id="@+id/textView17"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Aneesh" />
```

```
    <TextView
        android:id="@+id/textView18"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="95" />
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <TextView
        android:id="@+id/textView19"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="6" />
```

```
<TextView
```

```
    android:id="@+id/textView20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Atharva" />

<TextView
    android:id="@+id/textView21"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="88" />
</TableRow>

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

    <TextView
        android:id="@+id/textView22"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="7" />

    <TextView
        android:id="@+id/textView23"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="Pranay" />

    <TextView
        android:id="@+id/textView24"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="85" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<TextView
    android:id="@+id/textView25"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="8" />
```

```
<TextView
    android:id="@+id/textView26"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Adarsh" />
```

```
<TextView
    android:id="@+id/textView27"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="99" />
```

```
</TableRow>
```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<TextView
    android:id="@+id/textView28"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="9" />
```

```
<TextView
    android:id="@+id/textView29"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Sam" />
```

```
<TextView
    android:id="@+id/textView30"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="99" />
</TableRow>

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

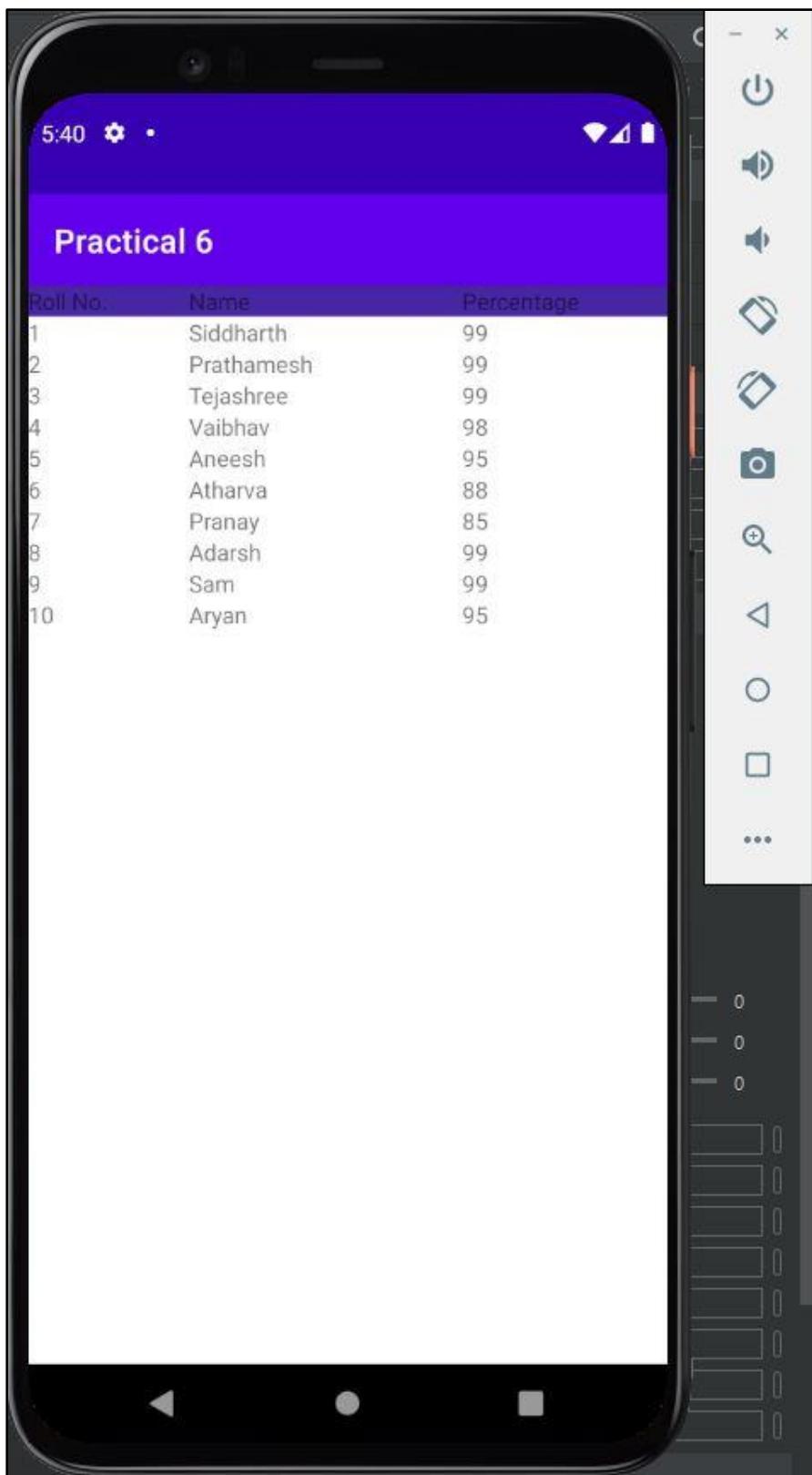
    <TextView
        android:id="@+id/textView31"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="10" />

    <TextView
        android:id="@+id/textView32"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Aryan" />

    <TextView
        android:id="@+id/textView33"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="95" />
</TableRow>

</TableLayout>
```

Output:



2. Write a program to display all the data types in object-oriented programming using Frame layout.

Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    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"
    tools:ignore="ExtraText">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="top|center"
        android:text="Datatypes "
        android:textAppearance="@style/TextAppearance.AppCompat.Large" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center|left"
        android:text="int"
        android:textAppearance="@style/TextAppearance.AppCompat.Large" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="float"
        android:textAppearance="@style/TextAppearance.AppCompat.Large" />

    <TextView
        android:id="@+id/textView3"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center|right"
    android:text="char"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

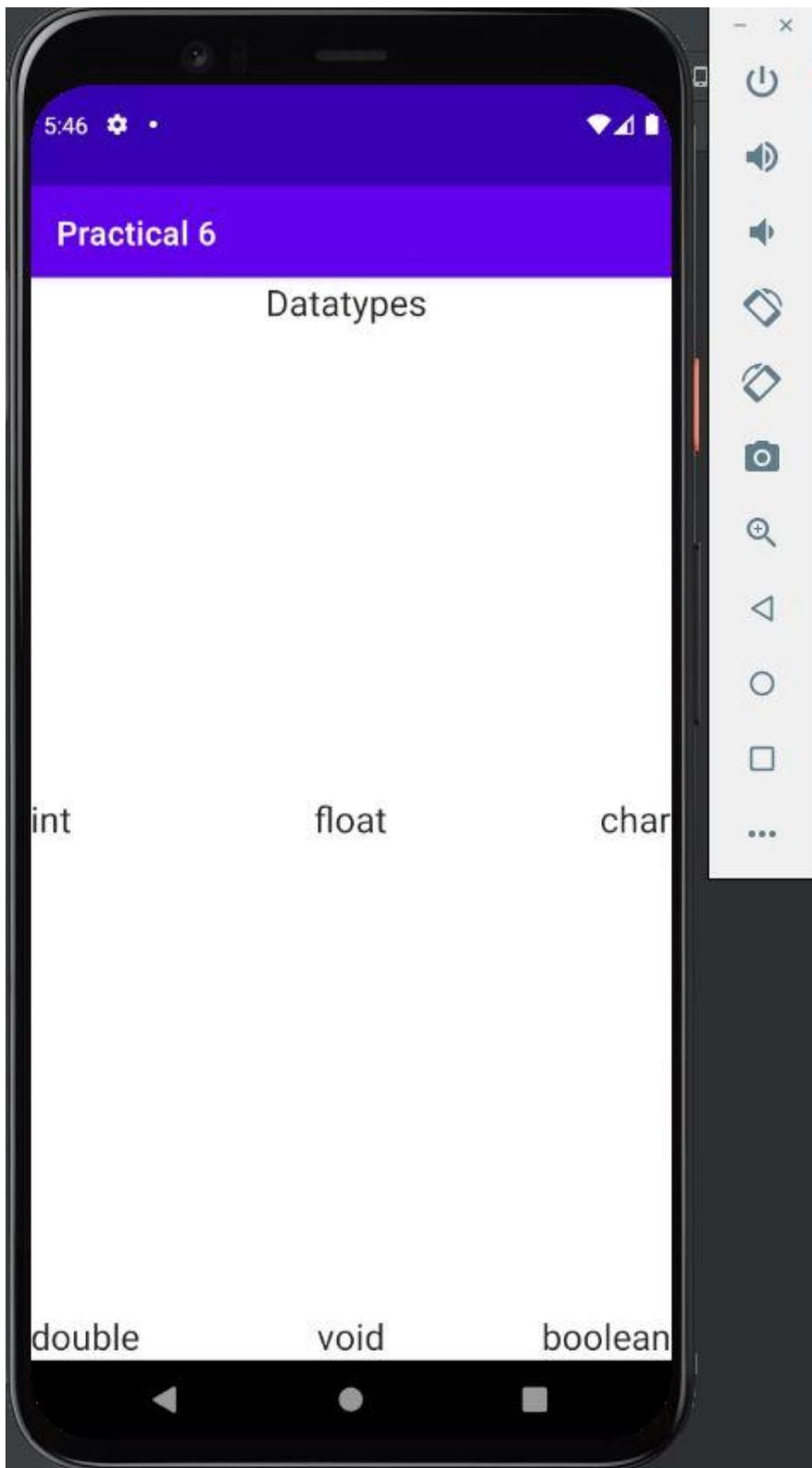
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|left"
    android:text="double"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|center"
    android:text="void"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|right"
    android:text="boolean"
    android:textAppearance="@style/TextAppearance.AppCompat.Large" />

</FrameLayout>
```

Output:



Practical 7

Develop a program to implement Text View and Edit Text.

Siddharth Revankar TYCO

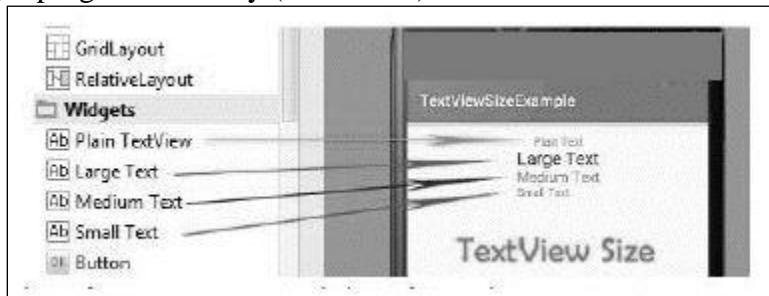
I. Practical Significance

In this practical, UI controls in android like Text view and edit view are studied. Wherein the UI controls can be developed, used and placed on the screen using different layout managers as per the problem statement requirements.

II. Minimum Theoretical Background

1. Text View:

In Android, Text View displays text to the user and optionally allows them to edit it programmatically. Text View is a complete text editor, however basic class is configured to not allow editing but we can edit it. View is the parent class of Text View Being a subclass of view the text view component can be used in your app's. GUI inside a View Group, or as the content view of an activity. We can create a Text View instance by declaring it inside a layout (XML file) or by instantiating it programmatically (Java Class).



2. Edit Text:

In Android, Edit Text is a standard entry widget in android apps. It is an overlay over Text View that configures itself to be editable. Edit Text is a subclass of Text View with text editing operations. Often use Edit Text in our applications in order to provide an input or text field, especially in forms. The simplest example of Edit Text is Login or Sign-in form. Text Fields in Android Studio are basically Edit Text.

Note: An Edit Text is simply a thin extension of a Text View. An Edit Text inherits all the properties of a Text View



III. Practical Related Questions

1. Which of these is not defined as a process state?

- a. Non-visible
- b. Visible
- c. Foreground
- d. Background

Ans. Non-visible is not defined as a process state

2. What is the name of the folder that contains the R.java file?

- a. src
- b. res
- c. bin
- d. gen

Ans. Gen folder contains the R.java file.

IV. Exercise

1. Write a program to accept username and password from the end user using Text View and Edit Text

Ans.

Activity_main.xml:

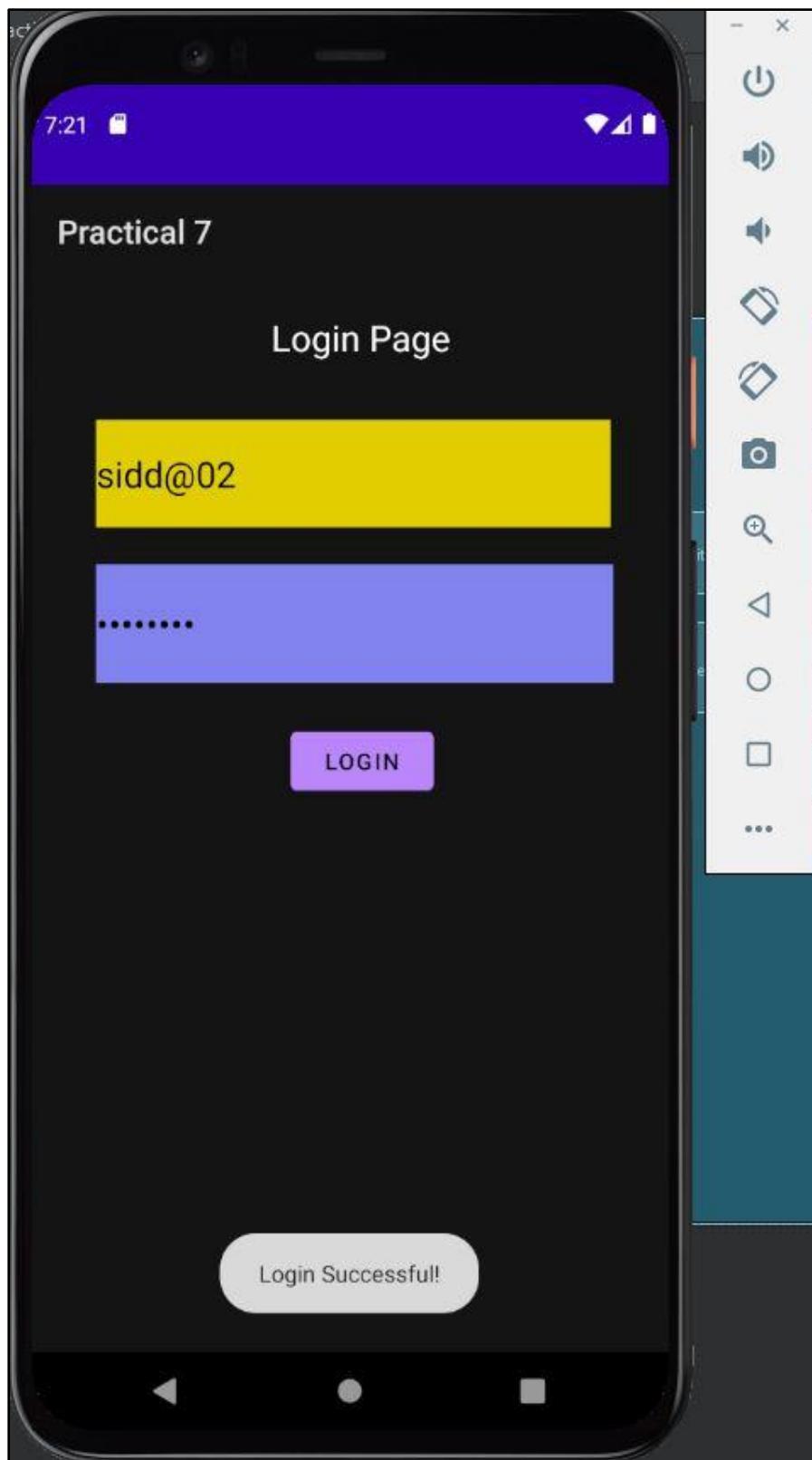
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"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

    <EditText
        android:id="@+id/editTextTextPersonName2"
        android:layout_width="317dp"
        android:layout_height="66dp"
        android:layout_marginStart="40dp"
        android:layout_marginLeft="40dp"
        android:layout_marginBottom="508dp"
        android:background="#E1CD0A"
        android:ems="10"
        android:hint="Username"
        android:inputType="textPersonName"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#000000"
        android:textColorHint="#000000"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="319dp"
        android:layout_height="73dp"
        android:layout_marginStart="40dp"
        android:layout_marginLeft="40dp"
        android:layout_marginBottom="412dp"
        android:background="#8383EF"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#000000"
        android:textColorHint="#000000"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="160dp"  
    android:layout_marginLeft="160dp"  
    android:layout_marginBottom="340dp"  
    android:text="Login"  
    android:onClick="onBtnClicked"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintStart_toStartOf="parent" />  
  
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="148dp"  
    android:layout_marginLeft="148dp"  
    android:layout_marginBottom="36dp"  
    android:text="Login Page"  
    android:textAppearance="@style/TextAppearance.AppCompat.Large"  
    android:textColor="#FFFFFF"  
    app:layout_constraintBottom_toTopOf="@+id/editTextTextPersonName2"  
    app:layout_constraintStart_toStartOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



2. Write a program to accept and display personal information of the student.

Ans.

Activity_main.xml:

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="140dp"
        android:layout_marginLeft="140dp"
        android:layout_marginBottom="47dp"
        android:text="Enter Details:"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#FFFFFF"
        app:layout_constraintBottom_toTopOf="@+id/editTextTextPersonName"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="100dp"
        android:layout_marginLeft="100dp"
        android:layout_marginBottom="12dp"
        android:ems="10"
        android:hint="Full Name"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toTopOf="@+id/editTextNumber"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/editTextNumber"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginBottom="448dp"
    android:ems="10"
    android:hint="Roll Number"
    android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginBottom="396dp"
    android:ems="10"
    android:hint="Blood Group"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<EditText
    android:id="@+id/editTextPhone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginBottom="340dp"
    android:ems="10"
    android:hint="Phone Number"
    android:inputType="phone"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="160dp"
    android:layout_marginLeft="160dp"
    android:layout_marginBottom="192dp"
    android:text="Submit"
    app:layout_constraintBottom_toBottomOf="parent"
```

```

    app:layout_constraintStart_toStartOf="parent" />

<EditText
    android:id="@+id/editTextTextPersonName3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginBottom="44dp"
    android:ems="10"
    android:hint="Address"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintStart_toStartOf="parent" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginBottom="48dp"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java:

```

package com.example.practical7;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

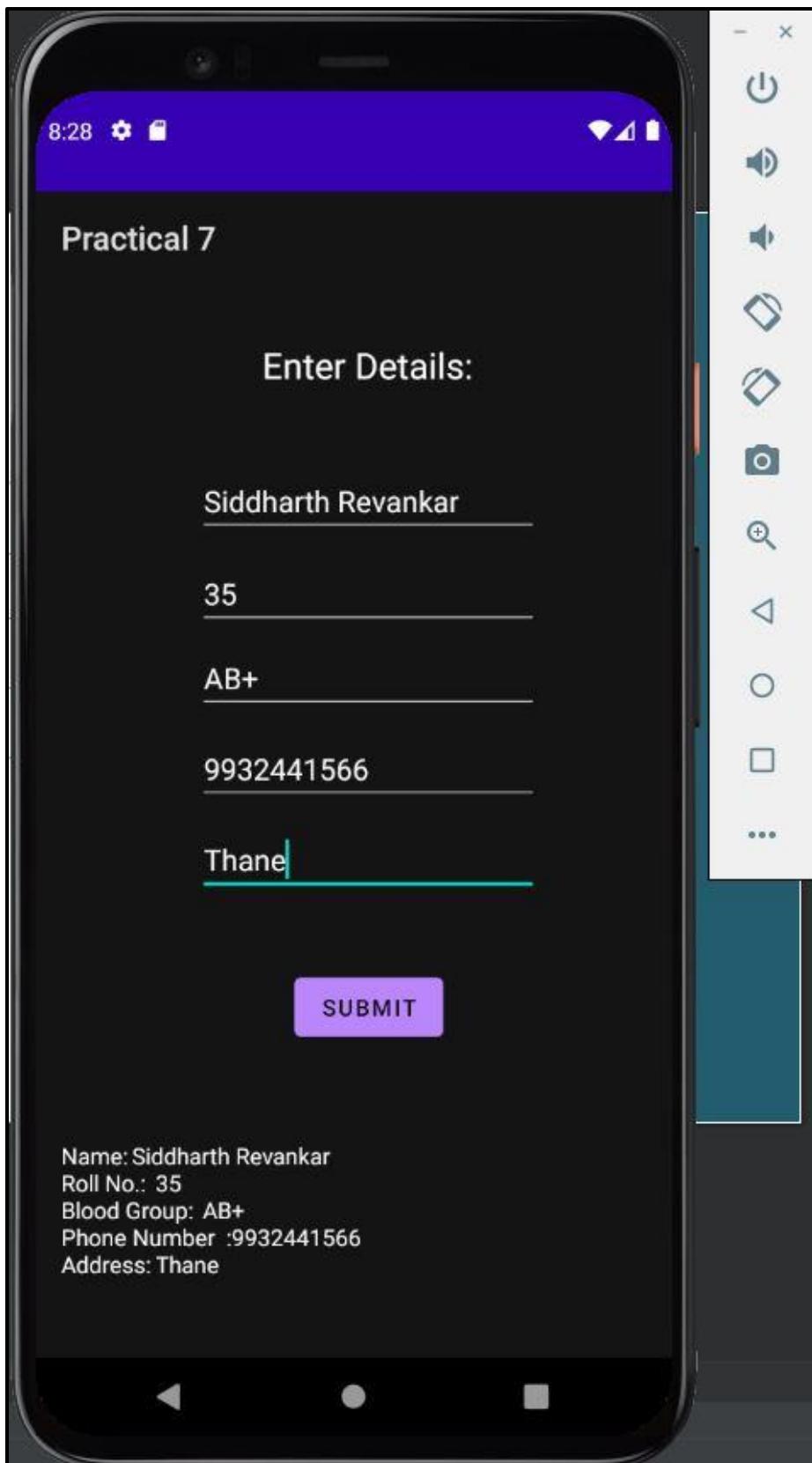
public class MainActivity extends AppCompatActivity {
    TextView result;

```

```
EditText eText,eText1,eText2,eText3,eText4;
Button btn;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    result = (TextView) findViewById(R.id.textView6);
    eText = (EditText) findViewById(R.id.editTextTextPersonName);
    eText1 = (EditText) findViewById(R.id.editTextNumber);
    eText2 = (EditText) findViewById(R.id.editTextTextPersonName2);
    eText3 = (EditText) findViewById(R.id.editTextPhone);
    eText4 = (EditText) findViewById(R.id.editTextTextPersonName3);
    btn = (Button) findViewById(R.id.button);
    btn.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            String name = eText.getText().toString();
            String roll = eText1.getText().toString();
            String bg = eText2.getText().toString();
            String phone = eText3.getText().toString();
            String add = eText4.getText().toString();
            result.setText("\nName:\t" + name + "\nRoll No.:\t" + roll + "\nBlood Group:\t" + bg + "\nPhone
Number\t:" + phone + "\nAddress:\t" + add);
        }
    });
}
}
```

Output:



Practical 8

Develop a program to implement Auto Complete Text View.

i. Practical Significance

In this practical, UI controls in android like Auto complete Text view is studied. Wherein the UI controls can be developed, used and placed on the screen using different layout managers as per the problem statement requirements.

ii. Minimum Theoretical Background

Auto Complete Text View:

Android Auto Complete Text View completes the word based on the reserved words, so no need to write all the characters of the word. Android Auto Complete Text View is a editable text field, it displays a list of suggestions in a drop down menu from which user can select only one suggestion or value. Android Auto Complete Text View is the subclass of Edit Text class. The Multi Auto Complete Text View is the subclass of Auto Complete Text View class. An editable text view that shows completion suggestions automatically while the user is typing. The list of suggestions is displayed in a dropdown menu from which the user can choose an item to replace the content of the edit box. The drop down can be dismissed at any time by pressing the back key or, if no item is selected in the drop down, by pressing the enter center key. The list of suggestions is obtained from a data adapter and appears only after a given number of characters defined by the threshold. Auto Complete Text View is a component used to show suggestions while writing in an editable text field. The suggestions list is shown in a drop-down menu from which a user can select the desired item. The list of suggestions is obtained from an adapter and it appears only after a number of characters that are specified in the threshold. To use an Auto Complete Threshold, it needs to be defined in the layout.



iii. Practical Related Questions

1. What does android:completionHint attribute in Auto Complete Text view do?

Ans – android:completionHint Defines the hint displayed in the drop down menu.

2. How to create AutoCompleteTextView field in XML?

Ans –

By adding following code in xml ,we can create AutoCompleteTextView field in xml:

```
<AutoCompleteTextView android:id="@+id/autoCompleteTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

iv. Exercise

1. Write a program to create a first display screen of any search engine using Auto

Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="275dp"
        android:layout_height="130dp"
        android:layout_x="65dp"
        android:layout_y="171dp"
        app:srcCompat="@drawable/logo" />
    <AutoCompleteTextView
        android:id="@+id/autoCompleteTextView"
        android:layout_width="277dp"
        android:layout_height="40dp"
        android:layout_x="65dp"
        android:layout_y="323dp"
        android:background="#63978E8E" />
    <ImageView
        android:id="@+id/imageView9"
        android:layout_width="match_parent"
        android:layout_height="84dp"
        android:layout_x="0dp"
        android:layout_y="510dp"
        android:scaleType="fitXY"
        app:srcCompat="@drawable/bottom" />
    <ImageView android:id="@+id/imageView10"
        android:layout_width="45dp"
        android:layout_height="39dp"
        android:layout_x="296dp"
        android:layout_y="324dp"
        app:srcCompat="@drawable/search" />
</AbsoluteLayout>
```

MainActivity.java:

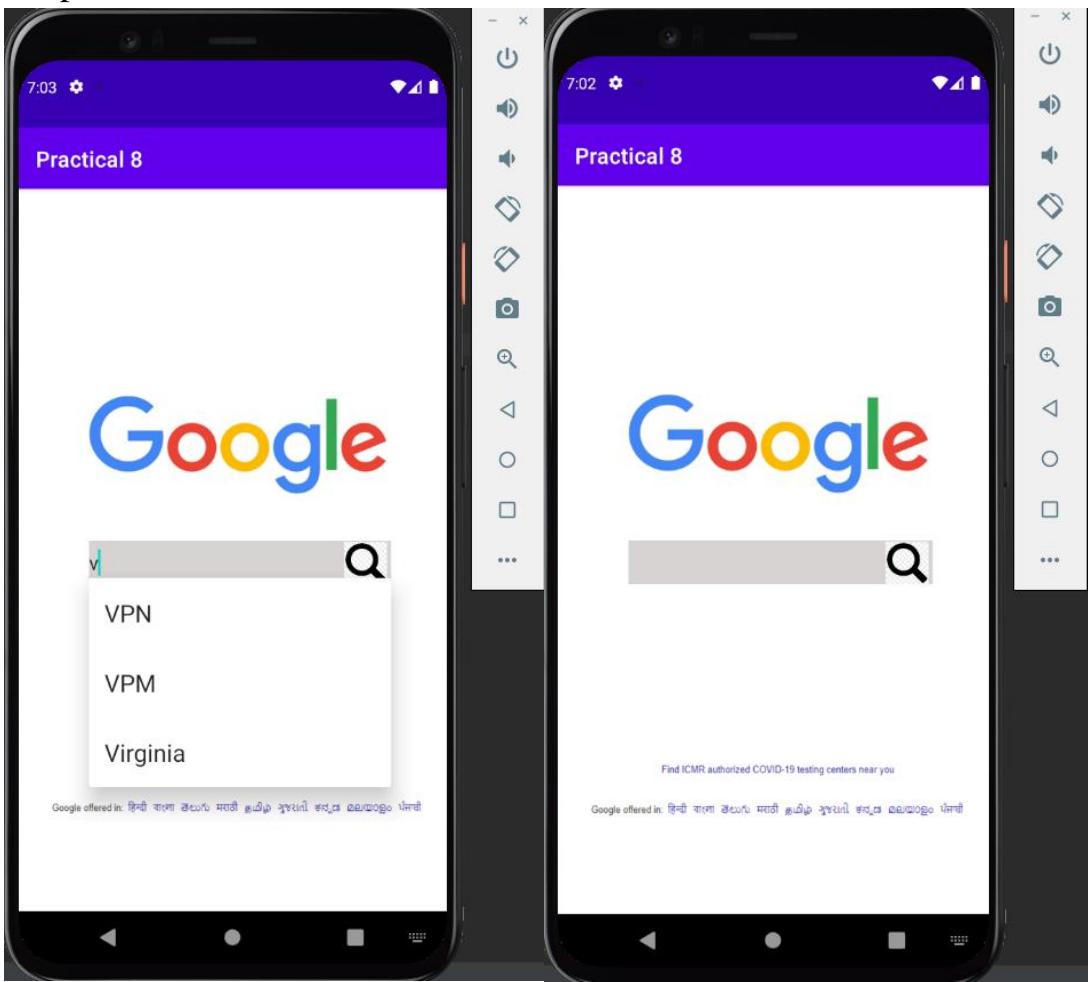
```
package com.example.practical8;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.*;

public class MainActivity extends AppCompatActivity {
    AutoCompleteTextView autocomplete;
    String[] arr = {"VPN", "VPM", "Virginia", "Google", "Google Chrome",
    "Youtube", "India", "Indie", "Google Duo"};
    @Override
    protected void onCreate(Bundle savedInstanceState)
    { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        autocomplete =
    (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>
    (this, android.R.layout.select_dialog_item, arr);
        autocomplete.setThreshold(1);
        autocomplete.setAdapter(adapter);
    }
}
```

Output:



2. Write a program to display all the subjects of sixth semester using AutoComplete Text View.

Ans.

Activity_main.xml:

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

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_x="30dp"
        android:layout_y="187dp"
        android:text="Select subject:"
```

```
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    />
```

```
<AutoCompleteTextView
    android:id="@+id/autoCompleteTextView"
    android:layout_width="221dp"
    android:layout_height="wrap_content"
    android:layout_x="150dp"
    android:layout_y="186dp" />
</AbsoluteLayout>
```

MainActivity.java:

```
package com.example.practical8;

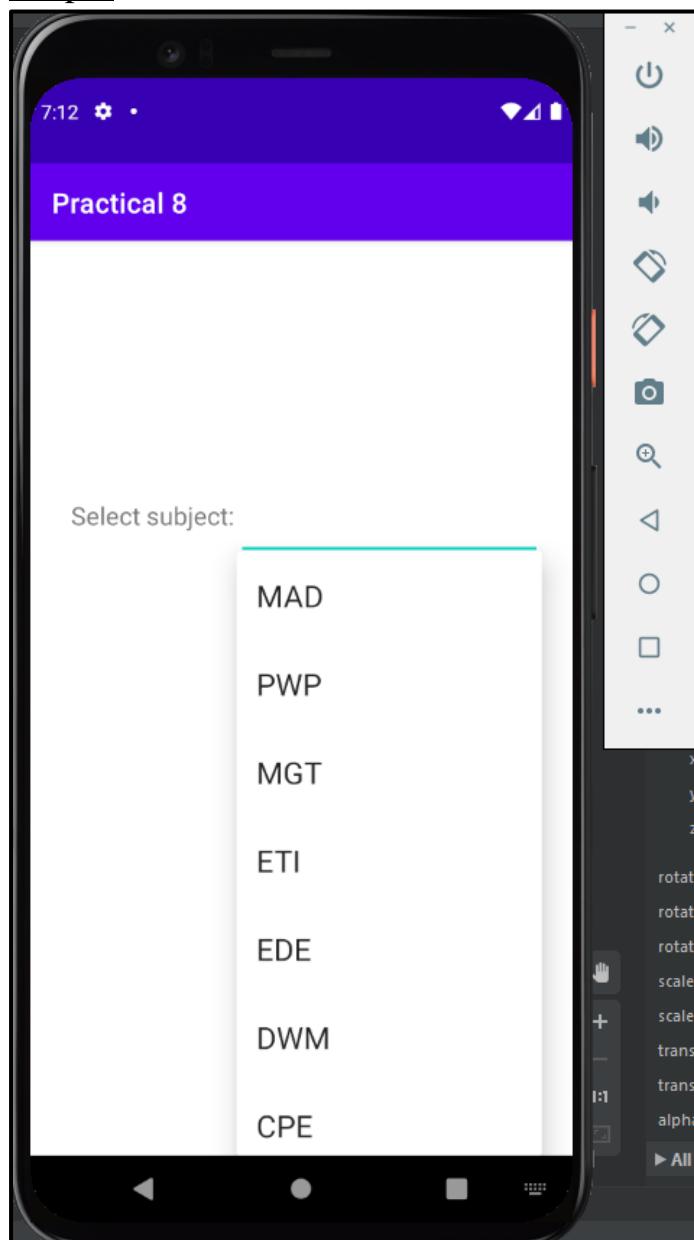
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View.*;
import android.view.*;
import android.widget.*;

public class MainActivity extends AppCompatActivity {
    AutoCompleteTextView autocomplete;
    String[] arr = {"MAD", "PWP", "MGT", "ETI", "EDE", "DWM", "CPE"};
    @SuppressLint("ClickableViewAccessibility")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        autocomplete =
            (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>
            (this, android.R.layout.select_dialog_item, arr);
        autocomplete.setThreshold(0);
        autocomplete.setAdapter(adapter);
```

```
autocomplete.setOnTouchListener(new OnTouchListener() {  
    @SuppressLint("ClickableViewAccessibility")  
    @Override  
    public boolean onTouch(View paramView, MotionEvent paramMotionEvent) {  
        autocomplete.showDropDown();  
        autocomplete.requestFocus();  
        return false;  
    }  
});  
}  
}
```

Output:



Practical 9

Develop a program to implement Button, Image Button and Toggle Button

Siddharth Revankar TYCO

I. Practical Significance

In this practical, UI controls in android like Buttons are studied. There are various types of buttons like Image button and toggle button which is studied.

II. Minimum Theoretical Background

1. Button:

In Android, Button represents a push button. The android.widget.Button is subclass of TextView class and Compound Button is the subclass of Button class. A Push buttons can be clicked, or pressed by the user to perform an action. There are different types of buttons used in android such as Compound Button, Toggle Button, Radio Button. Button is a subclass of TextView class and compound button is the subclass of Button class. On a button we can perform different actions or events like click event, pressed event, touch event etc. Android buttons are GUI components which are sensible to taps (clicks) by the user. When the user taps/clicks on button in an Android app, the app can respond to the click/tap. These buttons can be divided into two categories: the first is Buttons with text on, and second is buttons with an image on.



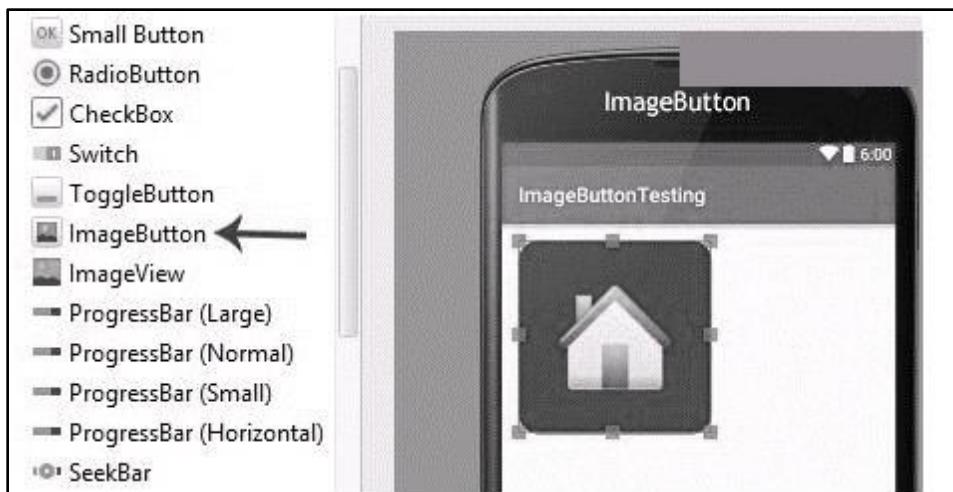
2. Types of Buttons:

Buttons can be divided into two categories the first is Buttons with text on, and second is buttons with an image on.

3. Image Button –

A button with images on can contain both an image and a text. Android buttons with images on are also called Image Button. In Android, Image Button is used to display a normal button with a custom image in a button. In simple words we can say, Image Button is a button with an image that can be pressed or clicked by the users. By default it looks like a normal button with the standard button background that changes the color during different button states. An image on the surface of a button is defined within a xml (i.e. layout) by using src attribute or within java class by using setImageResource() method. We can also set an image or custom drawable in the background of the image button . Image Button has all the properties of a normal button so you can easily perform any event like click or any other event which you can perform on a normal button.

Note: Standard button background image is displayed in the background of button whenever you create an image button. To remove that image, you can define your own background image in xml by using background attribute or in java class by using setBackground() method.



4. Toggle Button

A toggle button allows the user to change a setting between two states. You can add a basic toggle button to your layout With the Toggle Button object. If you need to Change a button's state yourself, you can use the Compound Button.setChecked() or Compound Button.toggle() method. To detect When the user activates the button or switch, create 21 Compound Button.OnCheckedChangeListener object and assign it to the button by calling setOnCheckedChangeListenerO. It is bene?cial if user have to change the setting between two states. It can be used to On/Off Sound, \Vi-Fi, Bluetooth etc. By default, the android Toggle Button Will be in OFF (Unchecked) state. We can Change the default state of Toggle Button by using androidzchecked attribute. In case, if we want to change the state of ToggleButton to ON (Checked), then we need to set androidzchecked = “true” in our XML layout file.



III. Practical Related Questions

1. Write a piece of code to set id of the button.

Ans.

Code: changeButton.setId(R.id.your_button_id);

2. How to add image to resources file?

Ans. To import image resources into your project, do the following:

1. Drag and drop your images directly onto the **Resource Manager** window in Android Studio. Alternatively, you can click the plus icon (+), choose **Import Drawables**, as shown in figure 3, and then select the files and folders that you want to import.
2. The **Import drawables** dialog appears.
3. The next screen shows a summary of the resources you're importing. When you're ready to import, click **Import**.

3. List four Android Toggle Button control attributes.

Ans. Four Toggle Button control attributes:

1. android:id
2. android:textOn
3. android:textOff
4. android:background

IV. Exercise

- 1. Write a program to create a toggle button to display ON / OFF Bluetooth on the display screen.**

Ans.

Activity_main.xml:

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Question 1"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#FFFFFF"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.176" />

    <ToggleButton
        android:id="@+id/toggleButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="160dp"
        android:layout_marginLeft="160dp"
        android:layout_marginBottom="436dp"
        android:text="Bluetooth"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

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

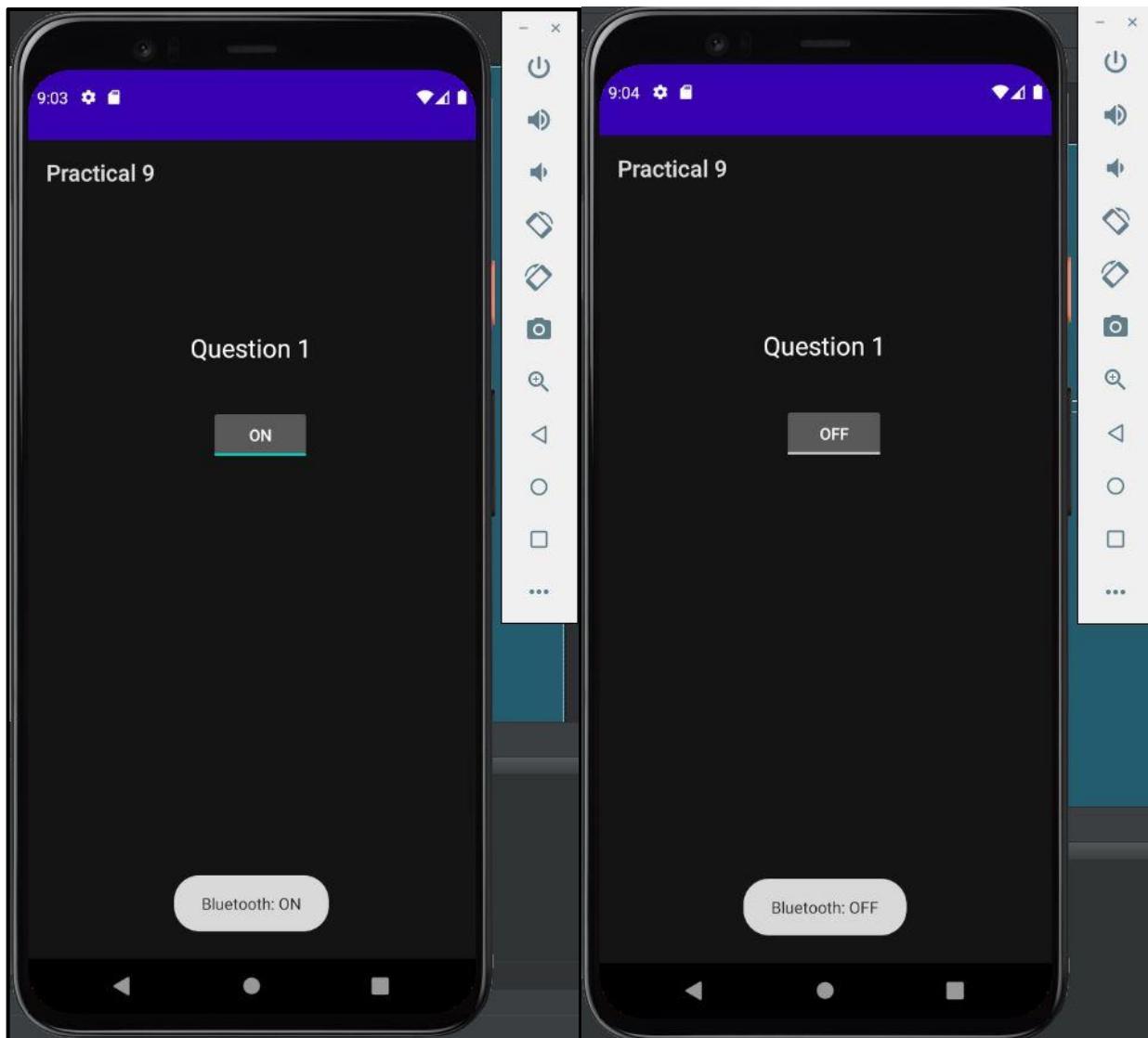
MainActivity.java:

```
package com.example.practical9;

import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Toast;
import android.widget.ToggleButton;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    ToggleButton toggleButton1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toggleButton1 = (ToggleButton) findViewById(R.id.toggleButton);
        toggleButton1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                StringBuilder result = new StringBuilder();
                result.append("Bluetooth: ").append(toggleButton1.getText());
                Toast.makeText(getApplicationContext(), result.toString(), Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Output:



2. Write a program to create a simple calculator

Ans.

Activity_main.xml:

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

```
tools:context="com.example.practical9.MainActivity">

<RelativeLayout
    android:id="@+id/relativeLayout"
    android:layout_width="368dp"
    android:layout_height="495dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <Button
        android:id="@+id	btn_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edText1"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginTop="60dp"
        android:onClick="PressOne"
        android:text="1"
        android:textSize="18sp"
        tools:ignore="OnClick" />

    <Button
        android:id="@+id	btn_0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id	btn_8"
        android:layout_toEndOf="@+id	btn_7"
        android:layout_toRightOf="@+id	btn_7"
        android:text="0"
        android:textSize="18sp" />

    <Button
        android:id="@+id	btn_9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id	btn_6"
        android:layout_toEndOf="@+id	btn_5"
        android:layout_toRightOf="@+id	btn_5"
```

```
    android:text="9"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_5"
    android:layout_toEndOf="@+id	btn_7"
    android:layout_toRightOf="@+id	btn_7"
    android:text="8"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_4"
    android:layout_alignStart="@+id	btn_4"
    android:layout_alignLeft="@+id	btn_4"
    android:text="7"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id	btn_5"
    android:layout_alignBottom="@+id	btn_5"
    android:layout_toEndOf="@+id	btn_5"
    android:layout_toRightOf="@+id	btn_5"
    android:text="6"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_2"
    android:layout_toEndOf="@+id	btn_4"
    android:layout_toRightOf="@+id	btn_4"
    android:text="5"
    android:textSize="18sp" />
```

```
<Button
    android:id="@+id	btn_4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_1"
    android:layout_alignStart="@+id	btn_1"
    android:layout_alignLeft="@+id	btn_1"
    android:text="4"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id	btn_2"
    android:layout_alignBottom="@+id	btn_2"
    android:layout_toEndOf="@+id	btn_2"
    android:layout_toRightOf="@+id	btn_2"
    android:text="3"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id	btn_1"
    android:layout_alignBottom="@+id	btn_1"
    android:layout_toEndOf="@+id	btn_1"
    android:layout_toRightOf="@+id	btn_1"
    android:text="2"
    android:textSize="18sp" />

<Button
    android:id="@+id	btn_Add"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id	btn_6"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:backgroundTint="@android:color/darker_gray"
    android:text="+"
    android:textColor="@android:color/background_light"
    android:textSize="18sp"
    tools:textColor="#FFFFFF" />
```

```
<Button
    android:id="@+id	btn_Sub"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_Add"
    android:layout_alignStart="@+id	btn_Add"
    android:layout_alignLeft="@+id	btn_Add"
    android:backgroundTint="@android:color/darker_gray"
    android:text="-"
    android:textColor="@android:color/background_light"
    android:textSize="18sp"
    tools:textColor="#FFFFFF" />

<Button
    android:id="@+id	btn_Mul"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_6"
    android:layout_alignStart="@+id	btn_Sub"
    android:layout_alignLeft="@+id	btn_Sub"
    android:backgroundTint="@android:color/darker_gray"
    android:text "*"
    android:textColor="@android:color/background_light"
    android:textSize="18sp"
    tools:textColor="#FFFFFF" />

<Button
    android:id="@+id	btn_Div"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id	btn_9"
    android:layout_alignStart="@+id	btn_Mul"
    android:layout_alignLeft="@+id	btn_Mul"
    android:backgroundTint="@android:color/darker_gray"
    android:text "/"
    android:textColor="@android:color/background_light"
    android:textSize="18sp"
    tools:textColor="#FFFFFF" />

<EditText
    android:id="@+id/edText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_gravity="end"
    android:layout_marginTop="22dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textAlignment="textEnd"
    android:textSize="24sp" />
```

```
<Button
    android:id="@+id/btn_calc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btn_0"
    android:layout_toEndOf="@+id/btn_0"
    android:layout_toRightOf="@+id/btn_0"
    android:backgroundTint="@android:color/holo_green_light"
    android:text ="="
    android:textColor="@android:color/background_light"
    android:textSize="18sp" />
```

```
<Button
    android:id="@+id/btn_dec"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btn_7"
    android:layout_toStartOf="@+id/btn_8"
    android:layout_toLeftOf="@+id/btn_8"
    android:text ="."
    android:textSize="18sp" />
```

```
<Button
    android:id="@+id/btn_clear"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btn_Div"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:backgroundTint="@android:color/holo_blue_dark"
    android:text="clear"
    android:textColor="@android:color/background_light"
```

```
    android:textSize="18sp" />

</RelativeLayout>

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="128dp"
    android:layout_marginLeft="128dp"
    android:layout_marginBottom="24dp"
    android:text="Simple Calculator"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toTopOf="@+id/relativeLayout"
    app:layout_constraintStart_toStartOf="parent" />
```

MainActivity.java:

```
package com.example.practical9;

import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    Button
btn_1,btn_2,btn_3,btn_4,btn_5,btn_6,btn_7,btn_8,btn_9,btn_0,btn_Add,btn_Sub,btn_Mul,btn_Div,btn
_calc,btn_dec,btn_clear;
    EditText ed1;

    float Value1, Value2;
    boolean mAddition, mSubtract, mMultiplication, mDivision ;

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

        btn_0 = (Button) findViewById(R.id.btn_0);
        btn_1 = (Button) findViewById(R.id.btn_1);
        btn_2 = (Button) findViewById(R.id.btn_2);
        btn_3 = (Button) findViewById(R.id.btn_3);
```

```
btn_4 = (Button) findViewById(R.id.btn_4);
btn_5 = (Button) findViewById(R.id.btn_5);
btn_6 = (Button) findViewById(R.id.btn_6);
btn_7 = (Button) findViewById(R.id.btn_7);
btn_8 = (Button) findViewById(R.id.btn_8);
btn_9 = (Button) findViewById(R.id.btn_9);
btn_Add = (Button) findViewById(R.id.btn_Add);
btn_Div = (Button) findViewById(R.id.btn_Div);
btn_Sub = (Button) findViewById(R.id.btn_Sub);
btn_Mul = (Button) findViewById(R.id.btn_Mul);
btn_calc = (Button) findViewById(R.id.btn_calc);
btn_dec = (Button) findViewById(R.id.btn_dec);
btn_clear = (Button) findViewById(R.id.btn_clear);
ed1 = (EditText) findViewById(R.id.edText1);

btn_0.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"0");
    }
});

btn_1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"1");
    }
});

btn_2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"2");
    }
});

btn_3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"3");
    }
});

btn_4.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"4");
    }
});

btn_5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"5");
    }
});

btn_6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"6");
    }
});

btn_7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"7");
    }
});

btn_8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"8");
    }
});

btn_9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText(ed1.getText()+"9");
    }
});

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

```
        ed1.setText(ed1.getText() + ".");
    }
});

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

        if (ed1 == null){
            ed1.setText("");
        }else {
            Value1 = Float.parseFloat(ed1.getText() + "");
            mAddition = true;
            ed1.setText(null);
        }
    }
});

btn_Sub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText() + "");
        mSubtract = true ;
        ed1.setText(null);
    }
});

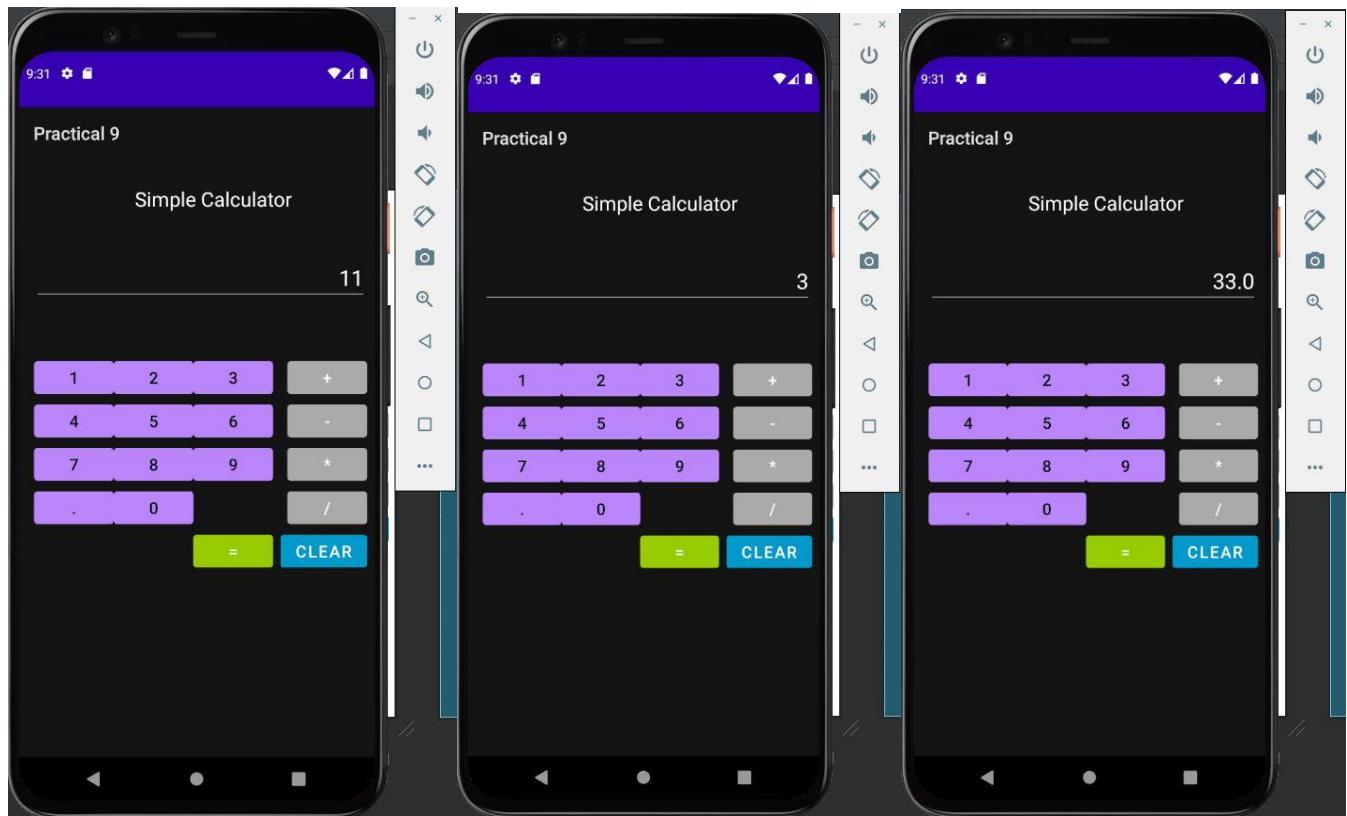
btn_Mul.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText() + "");
        mMultiplication = true ;
        ed1.setText(null);
    }
});

btn_Div.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value1 = Float.parseFloat(ed1.getText() + "");
        mDivision = true ;
        ed1.setText(null);
    }
});
```

```
btn_calc.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Value2 = Float.parseFloat(ed1.getText() + "");
        if (mAddition == true){
            ed1.setText(Value1 + Value2 + "");
            mAddition=false;
        }
        if (mSubtract == true){
            ed1.setText(Value1 - Value2 + "");
            mSubtract=false;
        }
        if (mMultiplication == true){
            ed1.setText(Value1 * Value2 + "");
            mMultiplication=false;
        }
        if (mDivision == true){
            ed1.setText(Value1 / Value2+"");
            mDivision=false;
        }
    });
});

btn_clear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        ed1.setText("");
    }
});
```

Output: Multiplication Demonstrated



Practical No.10

Develop a program to implement login window using above UI controls.

Siddharth Revankar TYCO

I. Practical Significance

In this practical, all the previous UI controls in android like Text View, Edit Text Buttons which are studied are implemented in this practical. Events are also handled on the android UI controls used in the practical.

VII. Minimum Theoretical Background

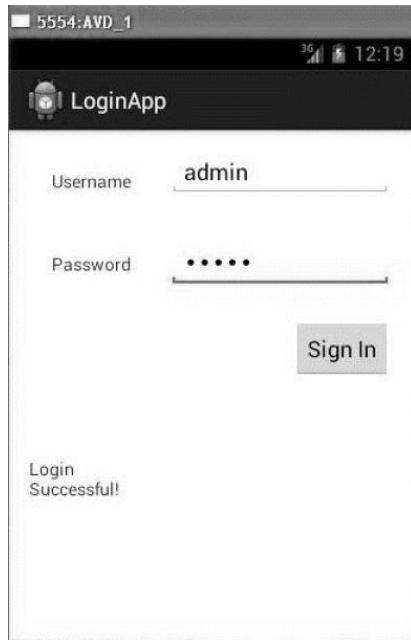
A login application is the screen asking your credentials to login to some particular application. You might have seen it when logging into facebook, twitter etc. Define two Text View asking username and password of the user. The password Text View must have input Type set to password. Its syntax is given below

```
<EditText  
    android:id = "@+id/editText2"  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
    android:inputType = "textPassword" />  
  
<EditText  
    android:id = "@+id/editText1"  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
/>
```

Define a button with login text and its onClick property.

```
<Button  
    android:id = "@+id/button1"  
    android:layout_width = "wrap_content"  
    android:layout_height = "wrap_content"  
    android:onClick = "login"  
    android:text = "@string/Login"  
/>
```

After that define the function mentioned in the onClick property in java file



IX. Practical Related Questions

1. Name the file in which respective XML components can be added.

Ans. In AndroidManifest.xml respective XML components can be added.

2. List all the UI components which can be used to develop login window.

Ans. UI Components which can be used are:

1. TextView
2. EditText
3. Button

X. Exercise

1. Write a program to create a login form for a social networking site.

Ans.

Activity_main.xml

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

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="70dp"
    android:layout_y="258dp"
```

```
    android:text="Enter User ID:"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintHorizontal_bias="0.256"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.293" />
```

```
<EditText  
    android:id="@+id/editTextTextPersonName"  
    android:layout_width="136dp"  
    android:layout_height="wrap_content"  
    android:layout_x="221dp"  
    android:layout_y="236dp"  
    android:ems="10"  
    android:inputType="textPersonName"  
    tools:layout_editor_absoluteX="211dp"  
    tools:layout_editor_absoluteY="209dp" />
```

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_x="71dp"  
    android:layout_y="320dp"  
    android:text="Enter Password:"  
    tools:layout_editor_absoluteX="79dp"  
    tools:layout_editor_absoluteY="254dp" />
```

```
<EditText  
    android:id="@+id/editTextTextPassword"  
    android:layout_width="141dp"  
    android:layout_height="wrap_content"  
    android:layout_x="220dp"  
    android:layout_y="300dp"  
    android:ems="10"  
    android:inputType="textPassword"  
    tools:layout_editor_absoluteX="211dp"  
    tools:layout_editor_absoluteY="254dp" />
```

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_x="150dp"
        android:layout_y="171dp"
        android:text="Login Page"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#FFFFFF"
        tools:layout_editor_absoluteX="153dp"
        tools:layout_editor_absoluteY="148dp" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="162dp"
    android:layout_y="388dp"
    android:onClick="show"
    android:text="Sign in" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="331dp"
    android:layout_height="64dp"
    android:layout_x="33dp"
    android:layout_y="106dp"
    app:srcCompat="@drawable/logo" />

</AbsoluteLayout>

```

MainActivity.java

```

package com.example.practical10;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

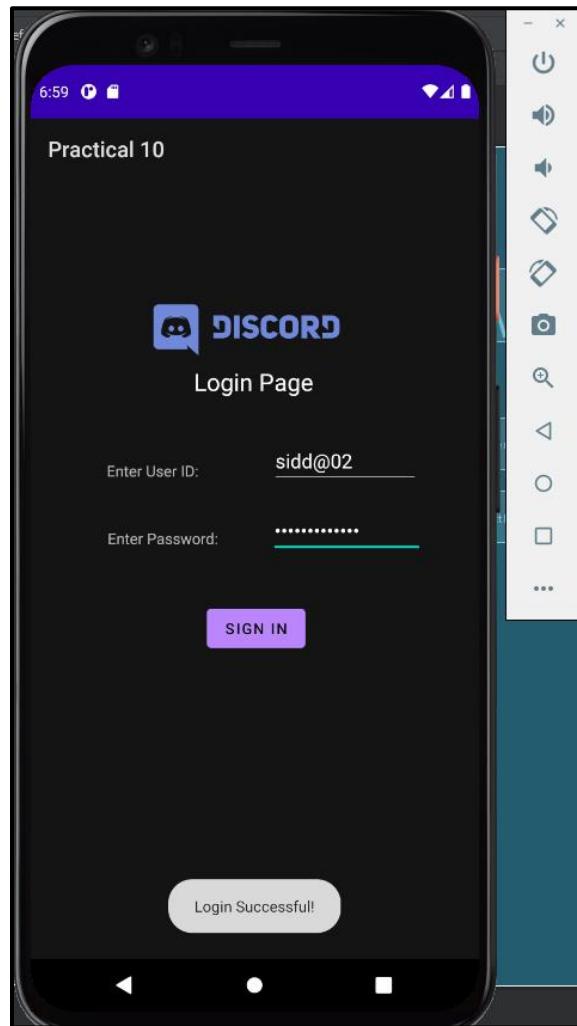
public class MainActivity extends AppCompatActivity {

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

```

```
    }
    public void show(View v){
        Toast.makeText(MainActivity.this,"Login Successful!",Toast.LENGTH_SHORT).show();
    }
}
```

Output:



2. Write a program to create a login form for student registration system

Ans.

Activity_main.xml

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

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="70dp"
    android:layout_y="258dp"
    android:text="Enter Name of Student:"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.256"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.293" />
```

```
<EditText
    android:id="@+id/editTextTextPersonName"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="221dp"
    android:layout_y="237dp"
    android:ems="10"
    android:inputType="textPersonName"
    tools:layout_editor_absoluteX="211dp"
    tools:layout_editor_absoluteY="209dp" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="70dp"
    android:layout_y="372dp"
    android:text="Enter Password:"
    tools:layout_editor_absoluteX="79dp"
    tools:layout_editor_absoluteY="254dp" />
```

```
<EditText
    android:id="@+id/editTextTextPassword"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="221dp"
    android:layout_y="366dp"
    android:ems="10"
    android:inputType="textPassword"
    tools:layout_editor_absoluteX="211dp"
```

```
tools:layout_editor_absoluteY="254dp" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="124dp"
    android:layout_y="175dp"
    android:text="Registration Form"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textColor="#FFFFFF"
    tools:layout_editor_absoluteX="153dp"
    tools:layout_editor_absoluteY="148dp" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="166dp"
    android:layout_y="569dp"
    android:onClick="show"
    android:text="Submit" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="143dp"
    android:layout_height="wrap_content"
    android:layout_x="70dp"
    android:layout_y="315dp"
    android:text="Enter Department:" />

<EditText
    android:id="@+id/editTextTextPersonName2"
    android:layout_width="142dp"
    android:layout_height="wrap_content"
    android:layout_x="221dp"
    android:layout_y="306dp"
    android:ems="10"
    android:inputType="textPersonName" />

<RadioGroup
    android:layout_width="124dp"
    android:layout_height="63dp"
```

```
    android:layout_x="222dp"
    android:layout_y="421dp">

    <RadioButton
        android:id="@+id radioButton"
        android:layout_width="65dp"
        android:layout_height="32dp"
        android:text="Male" />

    <RadioButton
        android:id="@+id radioButton2"
        android:layout_width="119dp"
        android:layout_height="25dp"
        android:text="Female" />
</RadioGroup>

<TextView
    android:id="@+id textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="69dp"
    android:layout_y="425dp"
    android:text="Select Gender:" />

<TextView
    android:id="@+id textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="69dp"
    android:layout_y="499dp"
    android:text="Date of Birth:" />

<EditText
    android:id="@+id editTextDate"
    android:layout_width="178dp"
    android:layout_height="wrap_content"
    android:layout_x="220dp"
    android:layout_y="497dp"
    android:ems="10"
    android:inputType="date" />

</AbsoluteLayout>
```

MainActivity.java

```
package com.example.practical10;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

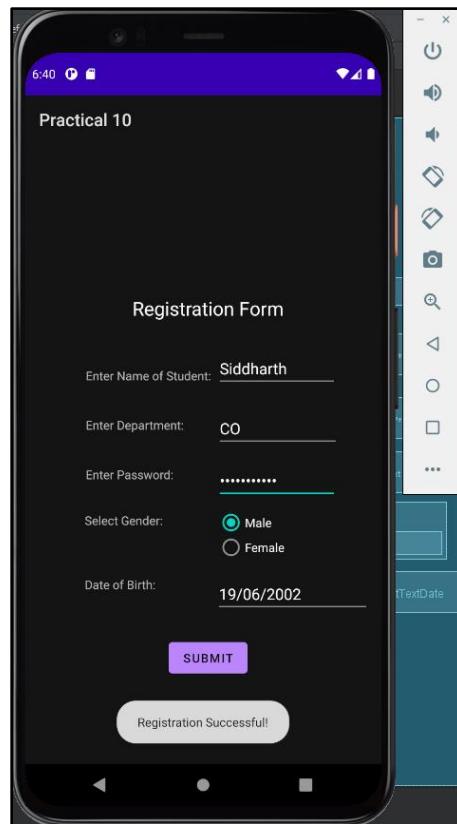
public class MainActivity extends AppCompatActivity {

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

    public void show(View v){
        Toast.makeText(MainActivity.this,"Registration
Successful!",Toast.LENGTH_SHORT).show();
    }

}
```

Output:



Practical No.11
Develop a program to implement checkbox
Siddharth Revankar TYCO

I. Practical Significance

Android CheckBox is a type of two state button either checked or unchecked. There can be a lot of usage of checkboxes. For example, it can be used to know the hobby of the user, activate/deactivate the specific action etc.

VII. Minimum Theoretical Background

Android Checkbox class is the subclass of Compound Button class.

Methods of Checkbox class

There are many inherited methods of View, Text View, and Button classes in the Checkbox Class. Some of them are as follows:

- public boolean isChecked(): Returns true if it is checked otherwise false.
- public void setChecked (boolean status) : Changes the state of the Checkbox.

Following figure shows different checkboxes



IX. Practical Related Questions

1. Name the different methods of Checkbox.

Ans. Methods of CheckBox:

1. public boolean isChecked(): Returns true if it is checked otherwise false.
2. public void setChecked(boolean status): Changes the state of the CheckBox.

2. List different attributes of Checkbox.

Ans. Attributes of Checkbox:

1. id
2. checked
3. gravity
4. text
5. textColor
6. textSize
7. textStyle
8. background
9. padding

3. Write xml tag to create a checkbox named “Android”.

Ans. <CheckBox

```
android:id="@+id/CheckBox1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Android"/>
```

X. Exercise

1. Write a program to show five checkboxes and toast selected checkboxes.

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout 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">
```

<CheckBox

```
    android:id="@+id/checkBox"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="28dp"
    android:layout_y="151dp"
    android:text="First" />
```

<CheckBox

```
    android:id="@+id/checkBox2"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
```

```
    android:layout_x="227dp"
    android:layout_y="151dp"
    android:text="Second" />
```

```
<CheckBox
    android:id="@+id/checkBox3"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="28dp"
    android:layout_y="222dp"
    android:text="Third" />
```

```
<CheckBox
    android:id="@+id/checkBox4"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="227dp"
    android:layout_y="221dp"
    android:text="Fourth" />
```

```
<CheckBox
    android:id="@+id/checkBox5"
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:layout_x="29dp"
    android:layout_y="291dp"
    android:text="Fifth" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="116dp"
    android:layout_y="85dp"
    android:text="Question 1"
    android:textAppearance="@style/TextAppearance.AppCompat.Display1" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="155dp"
    android:layout_y="361dp"
```

```
        android:onClick="show"
        android:text="Submit" />
</AbsoluteLayout>
```

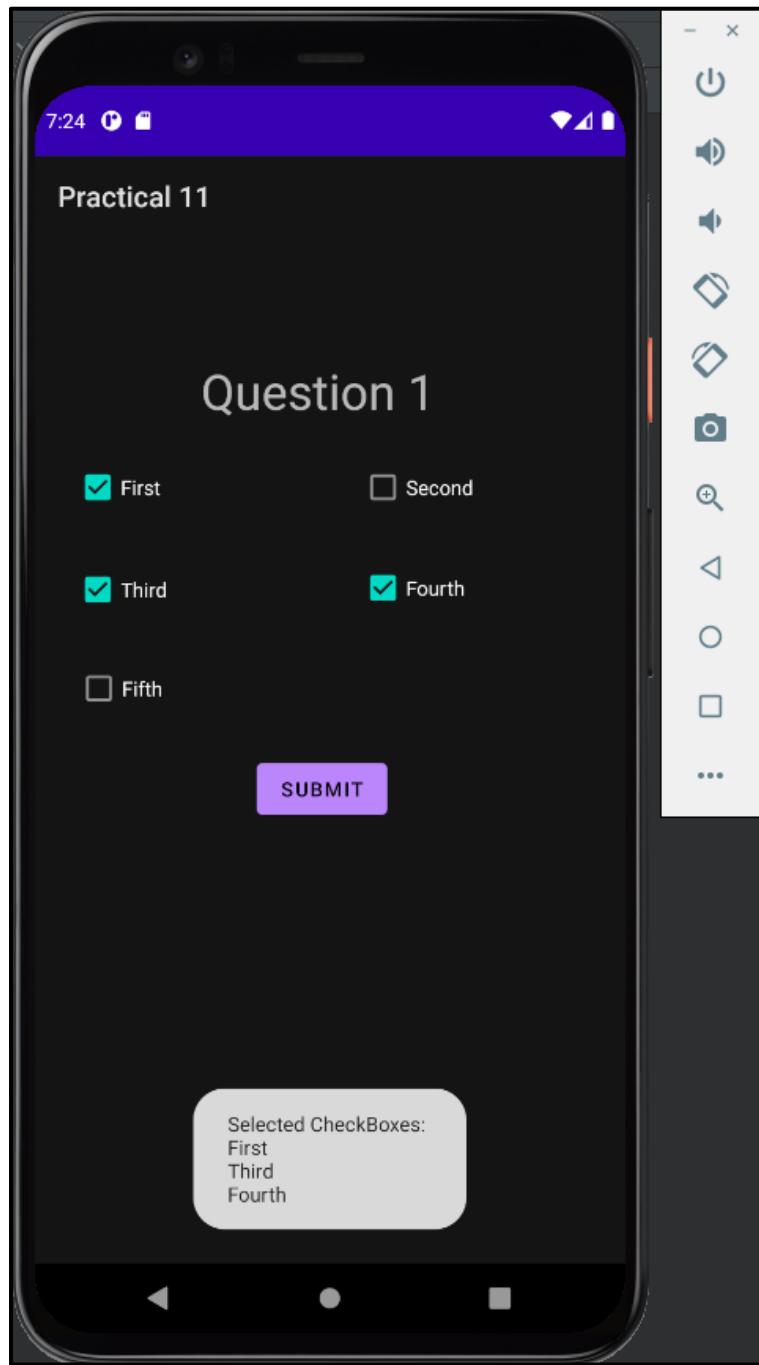
MainActivity.java

```
package com.example.practical11;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    CheckBox cb1,cb2,cb3,cb4,cb5;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void show(View v){
        cb1=(CheckBox)findViewById(R.id.checkBox);
        cb2=(CheckBox)findViewById(R.id.checkBox2);
        cb3=(CheckBox)findViewById(R.id.checkBox3);
        cb4=(CheckBox)findViewById(R.id.checkBox4);
        cb5=(CheckBox)findViewById(R.id.checkBox5);
        StringBuilder result=new StringBuilder();
        result.append("Selected CheckBoxes:");
        if(cb1.isChecked())result.append("\nFirst");
        if(cb2.isChecked())result.append("\nSecond");
        if(cb3.isChecked())result.append("\nThird");
        if(cb4.isChecked())result.append("\nFourth");
        if(cb5.isChecked())result.append("\nFifth");
        Toast.makeText(MainActivity.this,result.toString(),Toast.LENGTH_SHORT).show();
    }
}
```

Output:



Practical No.12

Aim: Develop a program to implement Radio Button and Radio Group.
Siddharth Revankar TYCO

I. Practical Significance

Radio Buttons are used when we need to select only one item from a list of presented items. If Radio Buttons are in Radio Group, when one Radio Button within a group is selected, all others are automatically deselected.

VII. Minimum Theoretical Background

Radio Button is generally used with *Radio Group*. **Radio Group** is a set of radio buttons, marking one radio button as checked makes all other radio buttons as unchecked. A radio button consists of two states – checked and unchecked. Clicking an unchecked button changes its state to “checked” state and “unchecked” for the previously selected radio button. To toggle a checked state to unchecked state, we need to choose another item.

Following are the important attributes related to Radio Group control.

1. **android:checkedButton** : This is the id of child radio button that should be checked by default within this radio group.
2. **android:orientation** : This property on the Radio group defines the orientation to position its child view consisting of Radio Buttons.

Following are the few methods of radio button:

1. **check(id)**: This sets the selection to the radio button whose identifier is passed in parameter. -1 is used as the selection identifier to clear the selection.
2. **clearCheck()** : It clears the selection. When the selection is cleared, no radio button in this group is selected and getCheckedRadioButtonId() returns null.
3. **getCheckedRadioButtonId()** : It returns the identifier of the selected radio button in this group. If its empty selection, the returned value is-1.
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



IX. Practical Related Questions

1. Write Xml tag to create a Radio button.

Ans.

```
<RadioButton  
    android:id="@+id/radioButton1"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:text="Radio Button 1"  
    android:layout_marginTop="20dp"  
    android:textSize="20dp" />
```

2. Write the purpose of Radio Button

Ans. Radio buttons are used when there is a list of two or more options that are mutually exclusive and the user must select exactly one choice. In other words, clicking a non-selected radio button will deselect whatever other button was previously selected in the list.

3. List different methods of Radio Button

Ans. Methods of Radio Button:

1. android:orientation
2. check(id)
3. clearCheck()
4. getCheckedRadioButtonId()
5. setOnCheckedChangeListener()

X. Exercise

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.

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:id="@+id/LinearLayout1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="100dp"
    android:text="Single Radio Buttons" />

<RadioButton
    android:id="@+id/radioButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_x="102dp"
    android:layout_y="121dp"
    android:text="Radio Button 1" />

<RadioButton
    android:id="@+id/radioButton2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_x="106dp"
    android:layout_y="185dp"
    android:text="Radio Button 2" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:layout_marginTop="20dp"
    android:background="#B8B894" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="50dp"
    android:text="RadioButtons inside RadioGroup" />

<RadioGroup
    android:id="@+id/RadioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
```

```
<RadioButton  
    android:id="@+id/RadioButton3"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Male" />
```

```
<RadioButton  
    android:id="@+id/RadioButton4"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Female" />
```

```
</RadioGroup>
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="30dp"  
    android:onClick="show"  
    android:layout_gravity="center_horizontal"  
    android:text="Show selected" />
```

```
</LinearLayout>
```

MainActivity.java

```
package com.example.prac12;
```

```
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.CheckBox;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.Toast;
```

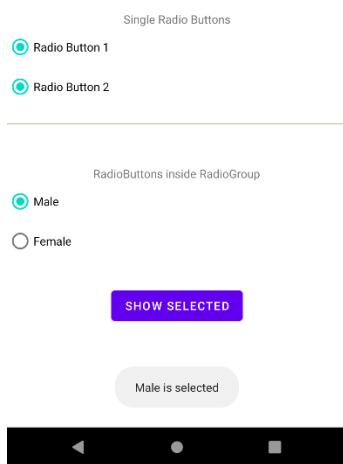
```
public class MainActivity extends AppCompatActivity {  
    RadioButton rb;  
    RadioGroup rg;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```

    }
    public void show(View v){
        rg=(RadioGroup)findViewById(R.id.RadioGroup1);
        int Id= rg.getCheckedRadioButtonId();
        rb = (RadioButton) findViewById(Id);
        Toast.makeText(MainActivity.this,rb.getText()+" is
selected",Toast.LENGTH_SHORT).show();
    }
}

```

Output:



Practical No.13

Develop a program to implement Progress Bar.

Siddharth Revankar TYCO

I. Practical Significance

Progress bars are used to show progress of a task. For example, when you are uploading or downloading something from the internet, it is better to show the progress of download/upload to the user. In android there is a class called Progress Dialog that allows you to create progress bar.

VII. Minimum Theoretical Background

A user interface element that indicates the progress of an operation. For a visual overview of the difference between determinate and indeterminate progress modes, see Progress & activity.

Display progress bars to a user in a non-interruptive way. Progress bar supports two modes to represent progress: determinate and indeterminate.

Indeterminate Progress

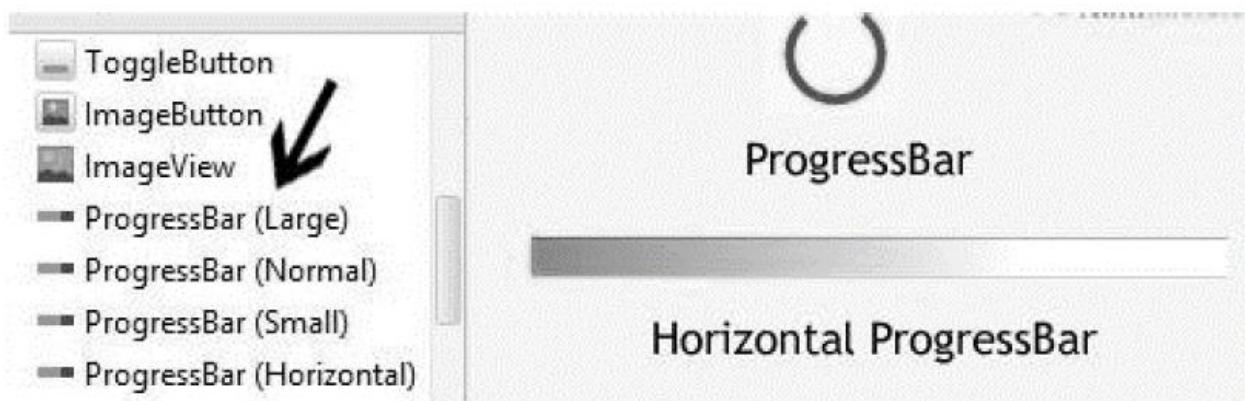
Use indeterminate mode for the progress bar when you do not know how long an operation will take. Indeterminate mode is the default for progress bar and shows a cyclic animation without a specific amount of progress indicated.

Determinate Progress

Use determinate mode for the progress bar when you want to show that a specific quantity of progress has occurred. For example, the percent remaining of a file being retrieved, the amount records in a batch written to database, or the percent remaining of an audio file that is playing.

ProgressDialog is a class that allows you to create progress bar. In order to do this, you need to instantiate an object of this class. Its syntax is.

```
ProgressDialog dialog=new ProgressDialog(this);
```



IX. Practical Related Questions

1. State different methods to update the percentage of progress displayed .

Ans.

You can update the percentage of progress displayed by using the setProgress(int) method, or by calling incrementProgressBy(int) to increase the current progress completed by a specified

amount. By default, the progress bar is full when the progress value reaches 100. You can adjust this default by setting the android:max attribute.

2. Write an Xml tag for the determinate progress bar

Ans. <ProgressBar

```
    android:id="@+id/determinateBar"
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:progress="25"/>
```

3. List different progress bar styles provided by the system.

Ans. Progress bar styles provided by the system include:

Widget.ProgressBar.Horizontal

Widget.ProgressBar.Small

Widget.ProgressBar.Large

Widget.ProgressBar.Inverse

Widget.ProgressBar.Small.Inverse

Widget.ProgressBar.Large.Inverse

X. Exercise

1. Write a program to display circular progress bar.

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout 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">
```

<ProgressBar

```
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_centerInParent="true"
    android:layout_x="104dp"
    android:layout_y="284dp"
    android:background="@drawable/circular_shape"
    android:indeterminate="false"
    android:max="100"
```

```
    android:progress="80"
    android:progressDrawable="@drawable/circular_progress_bar" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="81dp"
    android:layout_y="185dp"
    android:text="Progress Bar"
    android:textAppearance="@style/TextAppearance.AppCompat.Display2" />

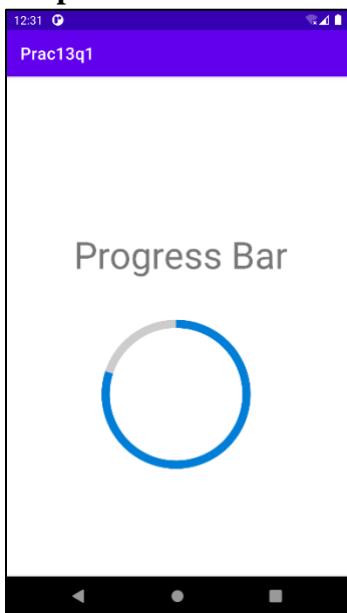
</AbsoluteLayout>
```

res/drawable/circular_shape.xml

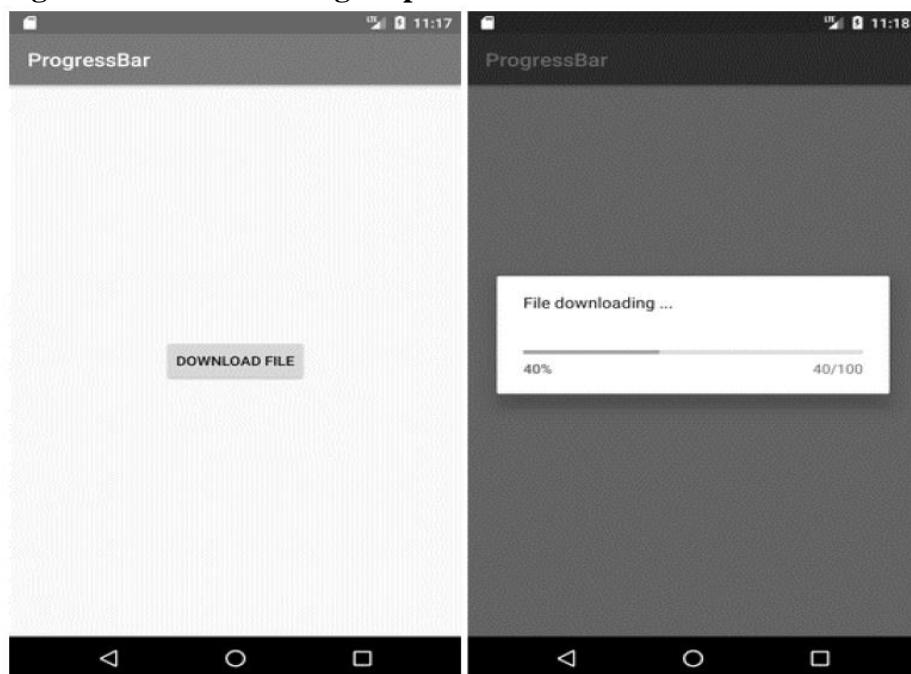
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android = "http://schemas.android.com/apk/res/android"
    android:innerRadiusRatio= "2.5"
    android:shape= "ring"
    android:thickness= "10dp"
    android:useLevel= "false" >
    <solid android:color= "#CCC" />
</shape>
```

res/drawable/circular_progress_bar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate 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= "10dp"
        android:useLevel= "true" >
        <gradient
            android:angle= "0"
            android:endColor= "#007DD6"
            android:startColor= "#007DD6"
            android:type= "sweep"
            android:useLevel= "false" />
    </shape>
</rotate>
```

Output:

2. Write a program to show following output.



Ans.

Activity_main.xml

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

```
tools:context=".MainActivity">

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="341dp"
    android:onClick="download"
    android:text="download file" />
</RelativeLayout>
```

MainActivity.java

```
package com.example.prac13q2;
import android.app.Activity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {
    Handler handler = new Handler();
    int status = 0;
    Button button;
    ProgressDialog progressdialog;

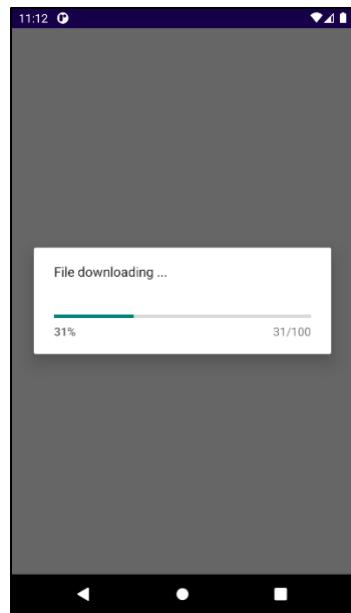
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CreateProgressDialog();
                ShowProgressDialog();
            }
        });
    }
}
```

```

public void CreateProgressDialog()
{
    progressdialog = new ProgressDialog(MainActivity.this);
    progressdialog.setIndeterminate(false);
    progressdialog.setMessage("File downloading ...");
    progressdialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
    progressdialog.setCancelable(true);
    progressdialog.setMax(100);
    progressdialog.show();
}
public void ShowProgressDialog()
{
    status = 0;
    new Thread(new Runnable() {
        @Override
        public void run() {
            while(status < 100){
                status +=1;
                try{
                    Thread.sleep(200);
                }catch(InterruptedException e){
                    e.printStackTrace();
                }
                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        progressdialog.setProgress(status);
                        if(status == 100){
                            progressdialog.dismiss();
                        }
                    }
                });
            }
        }).start();
    }
}

```

Output:



Practical No.14

Aim: Develop a program to implement List View, Grid View, Image View and Scroll View.

35 Siddharth Revankar TYCO

I. Practical Significance

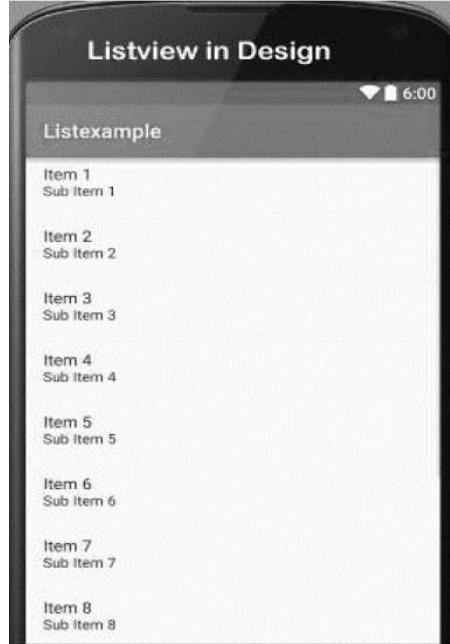
A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for *widgets*, which are used to create interactive UI components (buttons, text fields, etc.). The View Group subclass is the base class for *layouts*, which are invisible containers that hold other Views (or other View Groups) and define their layout properties.

VII. Minimum Theoretical Background

List View

List of scrollable items can be displayed in Android using List View. It helps you to displaying the data in the form of a scrollable list. Users can then select any list item by clicking on it. List View is default scrollable so we do not need to use scroll View or anything else with List View.

List View is widely used in android applications. A very common example of List View is your phone contact book, where you have a list of your contacts displayed in a List View and if you click on it then user information is displayed.



Grid View

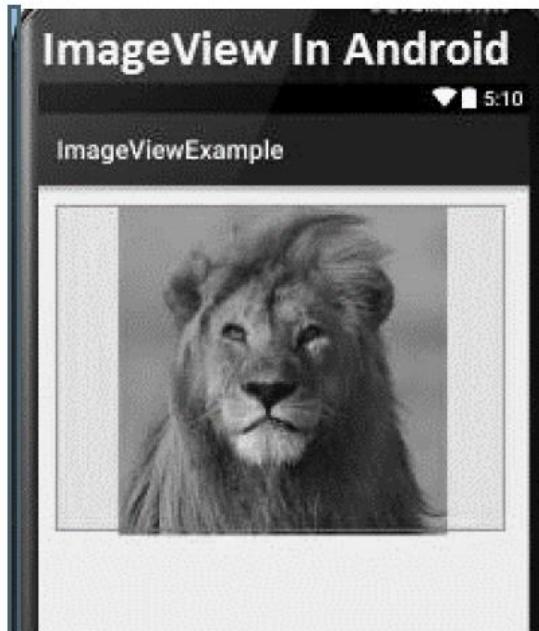
In android Grid View is a View group that display items in two dimensional scrolling grid (rows and columns), the grid items are not necessarily predetermined but they are automatically inserted

to the layout using a List Adapter. Users can then select any grid item by clicking on it. Grid View is default scrollable so we don't need to use Scroll View or anything else with Grid View.



Image View

In Android, Image View class is used to display an image file in application. Image file is easy to use but hard to master in Android, because of the various screen sizes in Android devices. An android is enriched With some of the best UI design widgets that allows us to build good looking and attractive UI based application.



Scroll View

In android scroll View can hold only one direct child. This means that, if you have complex layout With more Views(Buttons, Text Views or any other View) then you must enclose them inside another standard layout like Table Layout, Relative Layout or Linear Layout. You can specify

layout_width and layout_height to adjust Width and height of screen. You can specify height and Width in dp(density pixel) or px(pixel). Then after enclosing them in a standard layout, enclose the Whole layout in scroll View to make all the element or Views scrollable.



IX. Practical Related Questions

1. List all attributes of Image View.

Ans.

- a) android:scaleType
- b) android:src
- c) android:tint
- d) android:tintMode
- e) android:adjustViewBounds
- f) android:baseline

2. Write steps to add following string array to grid view.

```
static final String [] example= new String {"A", "B", "C", "D", "E"};
```

Ans.

Steps are as follow:

- 1.grid = (GridView) findViewById(R.id.gridView);
- 2.ArrayAdapter adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1, example);
- 3.grid.setAdapter(adapter);
- 4.grid.setOnItemClickListener code

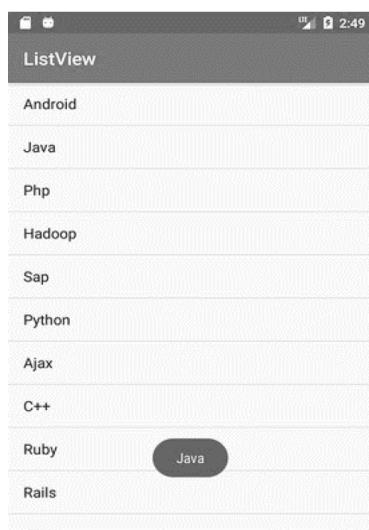
3. Describe android:stretchMode attribute of Grid view in detail.

Ans. It defines how columns should stretch to fill the available empty space, if any
spacingWidth: The spacing between each column is stretched.
columnWidth : Each column is stretched equally.

spacingWidthUniform : The spacing between each column is uniformly stretched.

X. Exercise

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



Ans.

Activity_main.xml

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

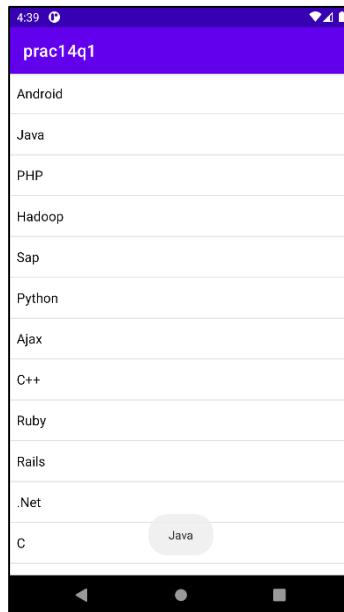
```
<ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac14q1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    ListView simpleList;
    ArrayAdapter<String> arrayAdapter;
    String countryList[] =
    {"Android","Java","PHP","Hadoop","Sap","Python","Ajax","C++","Ruby","Rails",".Net","C","MongoDB"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        simpleList = (ListView)findViewById(R.id.listview);
        arrayAdapter = new
        ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_item, countryList);
        simpleList.setAdapter(arrayAdapter);

        simpleList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long l)
            {
                String value=arrayAdapter.getItem(position);
                Toast.makeText(getApplicationContext(),value,Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Output:

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.

Ans.

Activity_main.xml

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

```
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/bg1" />
```

```
    <Button
        android:id="@+id/button"
```

```
    android:layout_width="312dp"
    android:layout_height="52dp"
    android:layout_marginTop="20dp"
    android:text="Change image"
    android:textSize="20sp"
    android:onClick="change"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

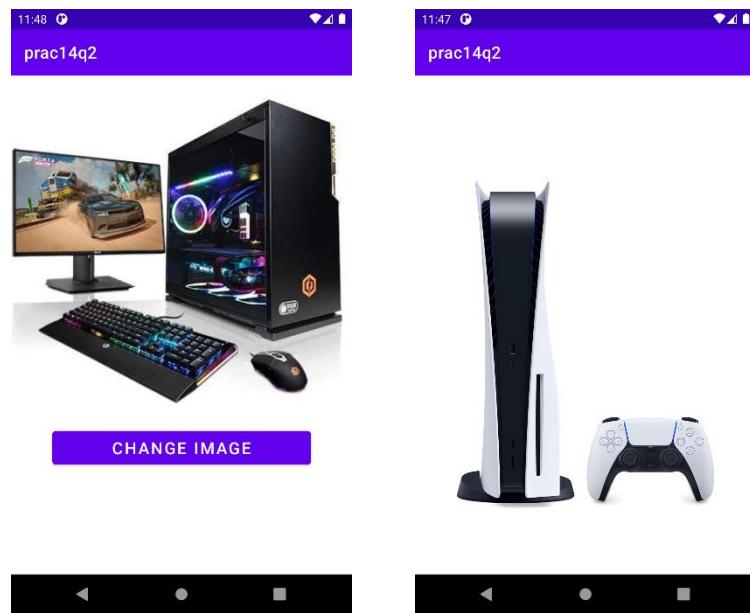
```
package com.example.prac14q2;

import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.*;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    private static ImageView img;
    private static Button btn;
    private int current_img;
    int[] images = {R.drawable.bg1,R.drawable.bg2};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void change(View v) {
        img = (ImageView)findViewById(R.id.imageView);
        btn = (Button)findViewById(R.id.button);
        current_img++;
        current_img = current_img % images.length;
        img.setImageResource(images[current_img]);
    }
}
```

Output:



3. Write a program to display 15 buttons using grid View.

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnWidth="110dp"
    android:numColumns="3"
    android:verticalSpacing="10dp"
    android:horizontalSpacing="10dp"
    android:stretchMode="columnWidth"
    android:gravity="center" />
```

ButtonAdapter.java

```
package com.example.prac14q3;
import android.content.Context;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.Toast;
```

```
public class ButtonAdapter extends BaseAdapter
{
    private Context mContext;
    private int btn_id;
    private int total_btns = 15;

    public ButtonAdapter(Context context) {
        this.mContext = context;
    }

    @Override
    public int getCount() {
        return total_btns;
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) {
        return 0;
    }

    @Override
    public View getView(final int i, View view, ViewGroup viewGroup)
    {
        Button btn;

        if (view == null) {
            btn = new Button(mContext);
            btn.setText("Button " + (++btn_id));
        } else {
            btn = (Button) view;
        }

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

```

```

        Toast.makeText(v.getContext(), "Button " + (i + 1), Toast.LENGTH_SHORT).show();
    }
});

return btn;
}
}

```

MainActivity.java

```

package com.example.prac14q3;

import androidx.appcompat.app.AppCompatActivity;

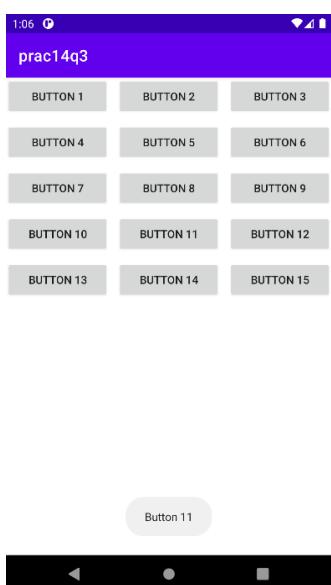
import android.os.Bundle;
import android.widget.GridView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        GridView gridview = (GridView) findViewById(R.id.gridview);
        gridview.setAdapter(new ButtonAdapter(this));
    }
}

```

Output:



4. Write a program to display a text View using vertical scroll View.

Ans.

Activity_main.xml

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

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textview"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </ScrollView>

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

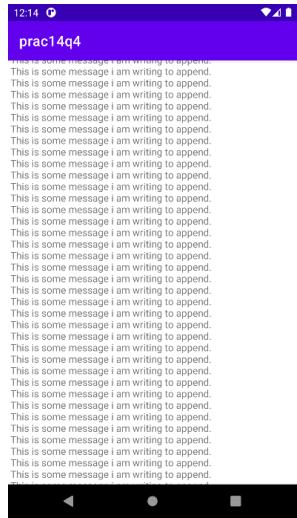
MainActivity.java

```
package com.example.prac14q4;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView msg = (TextView) findViewById(R.id.textview);
        StringBuilder stringBuilder = new StringBuilder();
        String someMessage = "This is some message i am writing to append.\n";
        for(int i = 0;i<100; i++){
            stringBuilder.append(someMessage);
        }
    }
}
```

```
        }  
        msg.setText(stringBuilder.toString());  
    }  
}
```

Output:



Practical No.15

Develop a program to implement Custom Toast Alert.

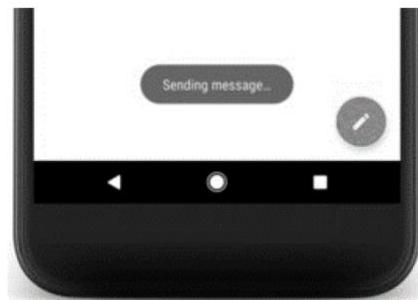
35 Siddharth Revankar TYCO

I. Practical Significance

Android Toast is a small message displayed on the screen, similar to a tool tip or other similar popup notification. Android Toast can be used to display information for the short period of time. A toast contains message to be displayed quickly and disappears after sometime. It provokes simple feedback about an operation in a small popup. A Toast is displayed on top of the main content of an activity. For example, navigating away from an email before you send it triggers a "Draft saved" toast to let you know that you can continue editing later.

VII. Minimum Theoretical Background

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains Visible and interactive. Toasts automatically disappear after a timeout. For example, clicking Send button on an email triggers a "Sending message..." toast



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.

IX. Practical Related Questions

1. 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?

Ans.

Predefined Constants-

- TOP
- BOTTOM

- LEFT
- RIGHT
- CENTER
- CENTER_HORIZONTAL
- CENTER_VERTICAL

2. List two constants of Toastclass.

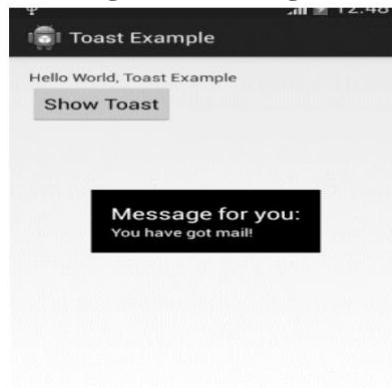
Ans.

The Two constants of Toast are :

- public static final int LENGTH_LONG
- public static final int LENGTH_SHORT.

X. Exercise

1. Write a program to display following toast message



Ans.

Activity_main.xml

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

```
<Button
    android:id="@+id/button"
    android:layout_width="132dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="84dp"
    android:layout_marginLeft="84dp"
    android:layout_marginTop="138dp"
    android:layout_marginEnd="195dp"
```

```
        android:layout_marginRight="195dp"
        android:layout_marginBottom="545dp"
        android:text="Show Toast"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

MainActivity.java

```
package com.example.prac15q1;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.widget.Button;
import android.widget.Toast;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    Button button;

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

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

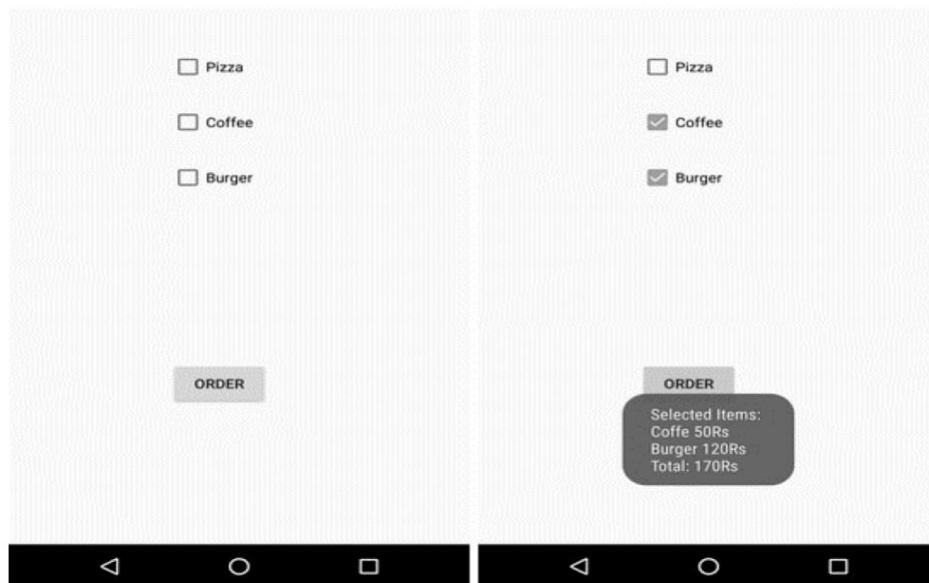
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {

                Toast.makeText(getApplicationContext(),
                        "Message for you : \nYou have got mail",Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Output:



2. Write a program to display three Checkboxes and one button named “Order “as shown below. Once you click on button it should toast different selected checkboxes along With items individual and total price.



Ans.

Activity_main.xml

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

```
        android:layout_height="match_parent"
        tools:context=".MainActivity">

<CheckBox
    android:id="@+id/checkBox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="144dp"
    android:layout_marginTop="68dp"
    android:text="Pizza"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<CheckBox
    android:id="@+id/checkBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="144dp"
    android:layout_marginTop="28dp"
    android:text="Coffee"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/checkBox" />

<CheckBox
    android:id="@+id/checkBox3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="144dp"
    android:layout_marginTop="28dp"
    android:text="Burger"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/checkBox2" />

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="144dp"
    android:layout_marginTop="184dp"
    android:text="Order"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/checkBox3" />  
  </androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.prac15q2;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.widget.Button;  
import android.widget.Toast;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.CheckBox;  
  
  
public class MainActivity extends AppCompatActivity {  
    CheckBox pizza, coffee, burger;  
    Button buttonOrder;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        addListenerOnButtonClick();  
    }  
    public void addListenerOnButtonClick(){  
  
        pizza=(CheckBox)findViewById(R.id.checkBox);  
        coffee=(CheckBox)findViewById(R.id.checkBox2);  
        burger=(CheckBox)findViewById(R.id.checkBox3);  
        buttonOrder=(Button)findViewById(R.id.button);  
        buttonOrder.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view)  
            {  
  
                int totalamount=0;  
                StringBuilder result=new StringBuilder();  
                result.append("Selected Items:");  
                if(pizza.isChecked()){  
                    result.append("\nPizza 100Rs");  
                    totalamount+=100;  
                }  
                if(co)
```

```

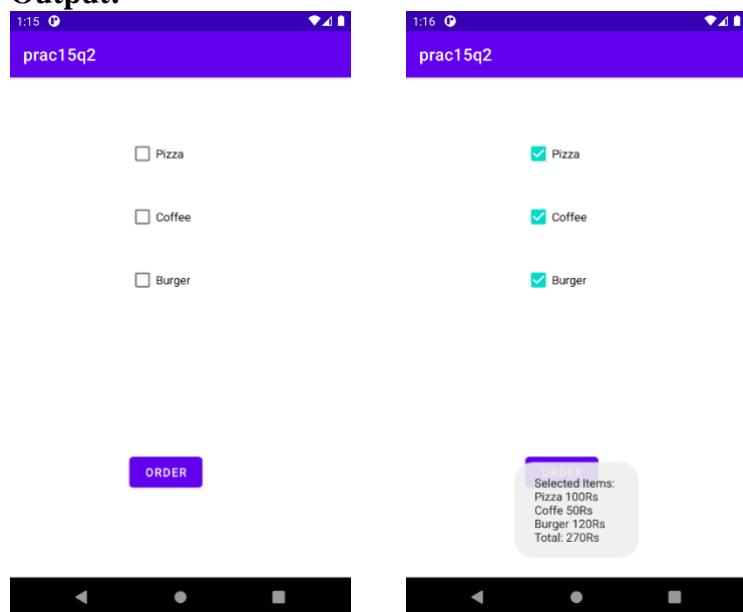
        ffee.isChecked()){
            result.append("\nCoffe 50Rs");
            totalamount+=50;
        }
        if(burger.isChecked()){
            result.append("\nBurger 120Rs");
            totalamount+=120;
        }
        result.append("\nTotal: "+totalamount+"Rs");
        Toast.makeText(getApplicationContext(), result.toString(),
        Toast.LENGTH_LONG).show();
    }

});
```

}

}

Output:



Practical No. 16

Develop a program to implement Date and Time Picker.

35 Siddharth Revankar TYCO

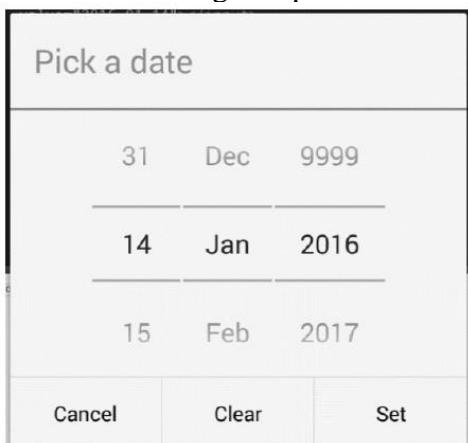
I. Practical Significance

Android provides controls for the user to pick a time or pick a date as ready-to-use dialogs. Each picker provides controls for selecting each part of the time (hour, minute, AM/PM) or date (month, day, year). Using these pickers helps ensure that your users can pick a time or date that is valid, formatted correctly, and adjusted to the user's locale.

VII. Minimum Theoretical Background

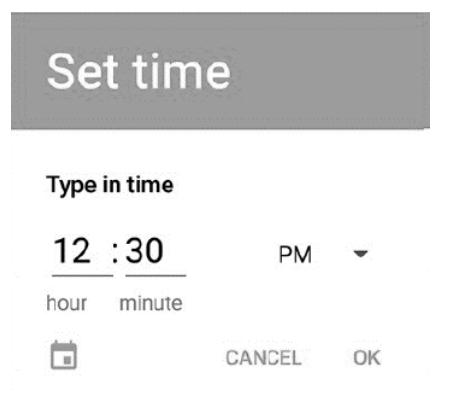
Date Picker:

Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.



Time Picker:

Android Time Picker allows you to select the time of day in either 24 hour or AM/PM mode. The time consists of hours, minutes and clock format. Android provides this functionality through TimePicker class. Following xml attribute is used to create time picker.



IX. Practical Related Questions

1. Write an xml Timepicker tag With all its attributes.

Ans.

```
<TimePicker android:id="@+id/timePicker1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

android:timePickerMode="clock" />

2. List and explain all methods of TimePiCker class

Ans. Methods of TimePiCker class

- **is24HourView():-** This method returns true if this is in 24 hour view else false
- **isEnabled():-** This method returns the enabled status for this view
- **setCurrentHour(Integer currentHour):-** This method sets the current hour
- **setCurrentMinute(Integer currentMinute):-** This method sets the current minute
- **setEnabled(boolean enabled):-** This method set the enabled state of this view
- **setIs24HourView(Boolean is24HourView):-** This method set whether in 24 hour or AM/PM mode
- **setOnTimeChangedListener(TimePicker.OnTimeChangedListener onTimeChangedListener):-** This method Set the callback that indicates the time has been adjusted by the user

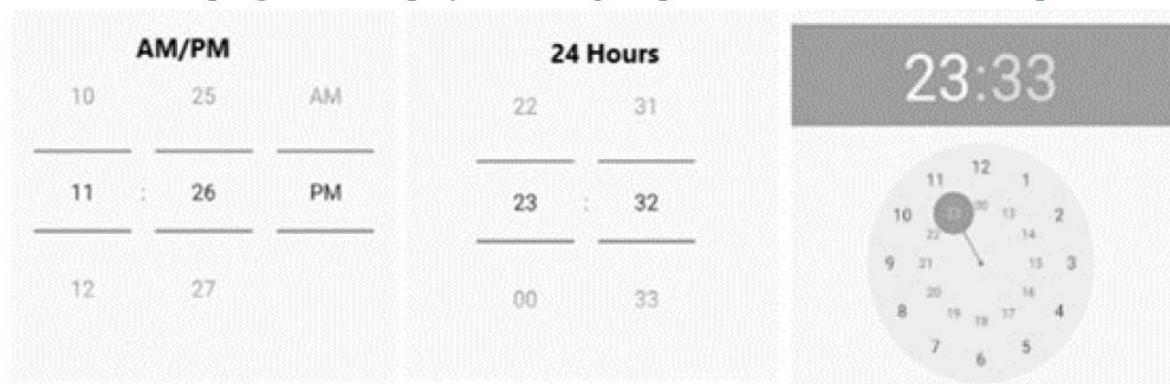
3. List and explain any five methods of DatePickerclass.

Ans. Methods of DatePicker class

- **getDayOfMonth():-** This method gets the selected day of month
- **getMonth():-** This method gets the selected month
- **getYear():-** This method gets the selected year
- **setMaxDate(long maxDate):-** This method sets the maximal date supported by this DatePicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone
- **setMinDate(long minDate):-** This method sets the minimal date supported by this NumberPicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone

X. Exercise

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



Ans.

a)

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Time"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.208" />

    <TimePicker
        android:id="@+id/time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:padding="20dp"
        android:timePickerMode="spinner"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.279"
        tools:ignore="MissingConstraints" />

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

MainActivity.java

```
package com.example.prac16q1;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
```

```

public class MainActivity extends AppCompatActivity {

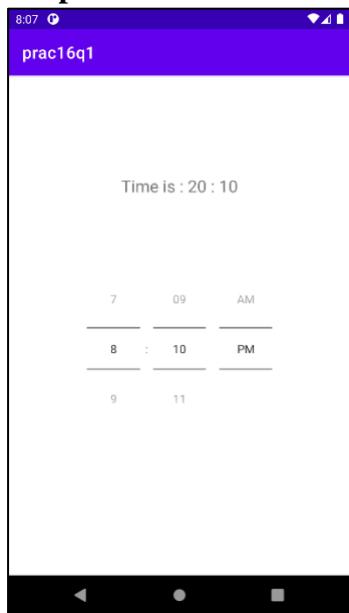
    TextView textView;
    TimePicker time;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        textView = (TextView) findViewById(R.id.textView);
        time = (TimePicker) findViewById(R.id.time);
        time.setIs24HourView(false);
        time.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {

            @Override
            public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
                textView.setText("Time is : " + hourOfDay + ":" + minute);
            }
        });
    }
}

```

Output:



b)

Activity_main.xml

```

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

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Time"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.208" />

    <TimePicker
        android:id="@+id/timePicker"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:padding="20dp"
        android:timePickerMode="spinner"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.279"
        tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.prac16q1b;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends AppCompatActivity {

    TextView textView;
    TimePicker timePicker;

```

```

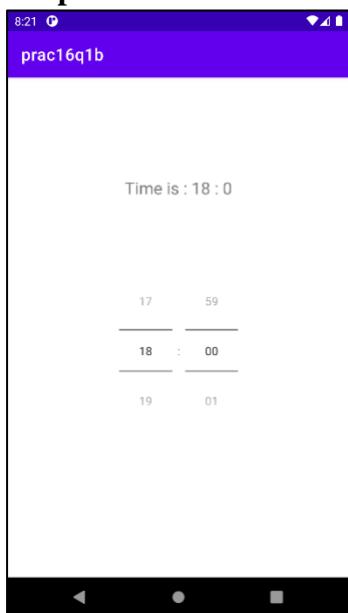
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textView = (TextView) findViewById(R.id.textView);
    timePicker = (TimePicker) findViewById(R.id.timePicker);
    timePicker.setIs24HourView(true);
    timePicker.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {

        @Override
        public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
            textView.setText("Time is : " + hourOfDay + ":" + minute);
        }
    });
}

```

Output:



c)

Activity_main.xml

```

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

```

```

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="124dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/timePicker" />

<TimePicker
    android:id="@+id/timePicker"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="64dp"
    app:layout_constraintTop_toTopOf="parent"
    tools:ignore="MissingConstraints"
    tools:layout_editor_absoluteX="0dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```

package com.example.prac16q1c;
import androidx.appcompat.app.AppCompatActivity;

import android.app.TimePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends AppCompatActivity {

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

        textView= (TextView)findViewById(R.id.textView);
        timePicker=(TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(false);
        textView.setText(getCurrentTime());
    }
}

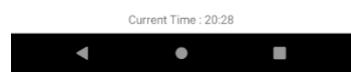
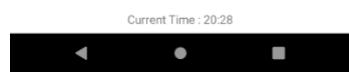
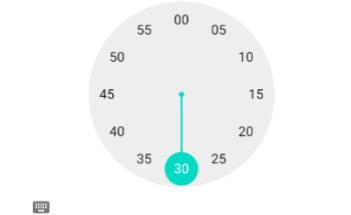
```

```

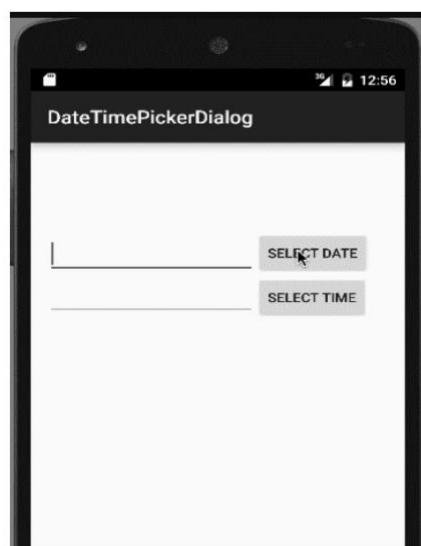
        }
    public String getCurrentTime()
    {
        String currentTime="Current Time : "+
timePicker.getCurrentHour()+":"+timePicker.getCurrentMinute();
        return currentTime;
    }
}

```

Output:



2. Write a program to display following output. Select and display date and time on Click of “select date”, “select time” buttons respectively.



Ans.

Activity_main.xml

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

```
<EditText
    android:id="@+id/textdate"
    android:layout_width="150sp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.146"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.204" />
```

```
<Button
    android:id="@+id/buttondate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/textdate"
    android:text="SELECT DATE"
    app:backgroundTint="#00BCD4"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.863"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.204" />
```

```
<EditText
    android:id="@+id/texttime"
    android:layout_width="150sp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.146"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textdate"
    app:layout_constraintVertical_bias="0.155" />

<Button
    android:id="@+id/buttontime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/buttondate"
    android:text="SELECT TIME"
    app:backgroundTint="#00BCD4"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.86"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/buttondate"
    app:layout_constraintVertical_bias="0.156" />

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

MainActivity.java

```
package com.example.prac16q2;

import androidx.appcompat.app.AppCompatActivity;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.TimePicker;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button buttondate, buttontime;
    int day, month, year, hour, minute;
    EditText textdate, texttime;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

buttondate=(Button)findViewById(R.id.buttondate);
butontime=(Button)findViewById(R.id.butontime);
textdate=(EditText)findViewById(R.id.textdate);
texttime=(EditText)findViewById(R.id.texttime);

buttondate.setOnClickListener(this);
butontime.setOnClickListener(this);

}

public void onClick(View view){
    if(view == buttondate)
    {
        final Calendar c=Calendar.getInstance();
        year = c.get(Calendar.YEAR);
        month = c.get(Calendar.MONTH);
        day = c.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog datePickerDialog = new DatePickerDialog(this,
                new DatePickerDialog.OnDateSetListener() {

            @Override
            public void onDateSet(DatePicker view, int year,
                    int monthOfYear, int dayOfMonth) {

                textdate.setText(dayOfMonth + "-" + (monthOfYear + 1) + "-" + year);

            }
        }, year, month, day);
        datePickerDialog.show();
    }

    if (view == butontime) {

        // Get Current Time
        final Calendar c = Calendar.getInstance();
        hour = c.get(Calendar.HOUR_OF_DAY);
        minute = c.get(Calendar.MINUTE);

        // Launch Time Picker Dialog
        TimePickerDialog timePickerDialog = new TimePickerDialog(this,
                new TimePickerDialog.OnTimeSetListener() {

            @Override

```

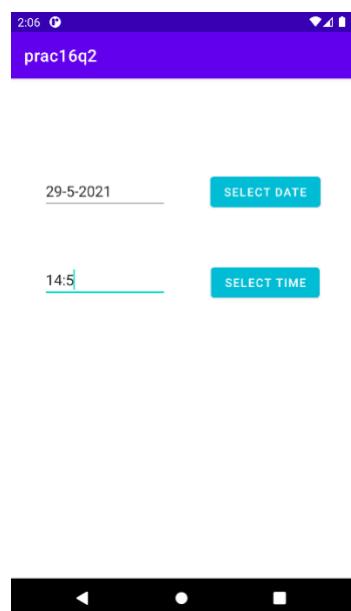
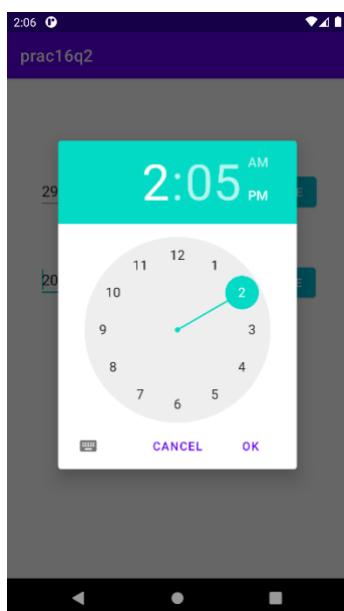
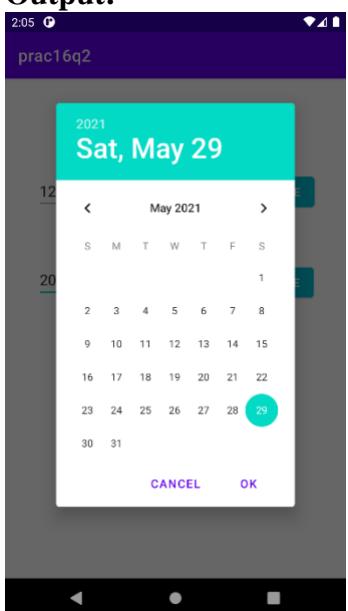
```

        public void onTimeSet(TimePicker view, int hourOfDay,
                              int minute) {
            texttime.setText(hourOfDay + ":" + minute);
        }
    }, hour, minute, false);
timePickerDialog.show();

}
}
}
}

```

Output:



Practical No. 17

Develop a program to create an activity

35 Siddharth Revankar TYCO

I. Practical Significance

An activity represents a single screen with a user interface. For example, an email application might have one activity that shows a list of new emails, another activity to compose an email, and one for reading emails.

VII. Minimum Theoretical Background

An activity is the single screen in android. It is like window or frame of Java. By the help of activity, you can place all your UI components or widgets in a single screen. If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched. As in C, C++ or Java programming language program starts from main () function, android system initiates its program within an Activity starting with a call on onCreate() method. Android Activity class is the subclass of ContextThemeWrapper class. An activity class loads all the UI component using the XML file available in res/layout folder of the project. Following statement loads UI components from res/layout/activity_main.xmlfile:

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

To write our own activity the new activity must be the derived from Activity class as given below

```
public class MainActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        /*Code goes here*/
    }
    /** Called when the activity is about to
     * become visible. */
    @Override
    protected void onStart() { super.onStart();
        /*Code goes here*/
    }
    /** Called when the activity has
     * become visible. */
    @Override
    protected void onResume() { super.onResume();
        /*Code goeshere*/
    }
    /** Called when another activity is
     * taking focus. */
    @Override
    protected void
    onPause() {
        super.onPause();
        /*Code goes here*/
    }
    /** Called when the activity is no
```

```

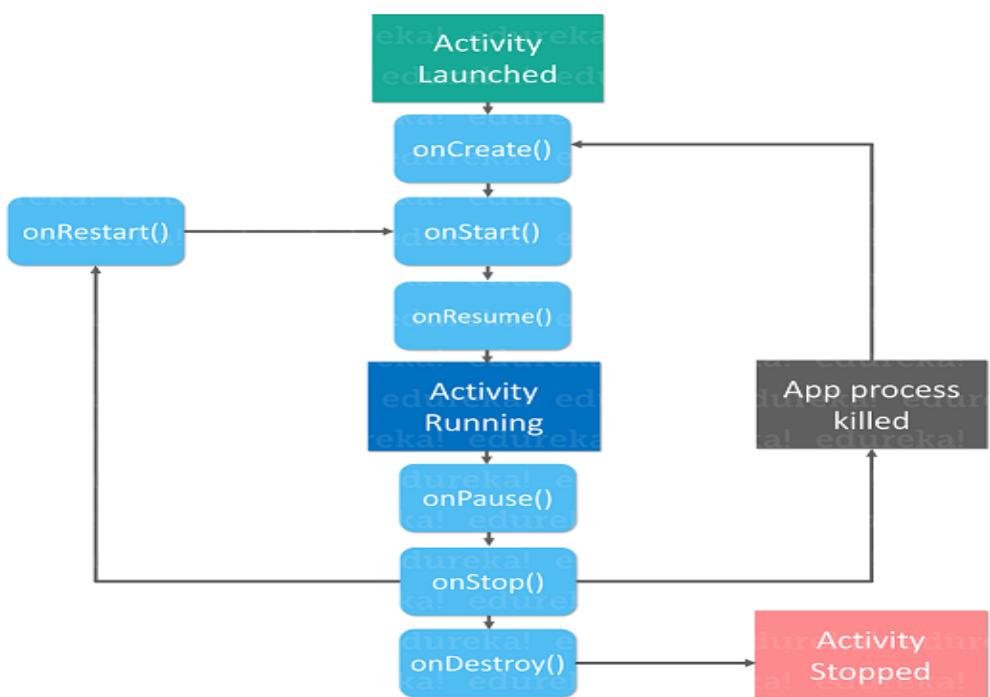
longer visible. */ @Override
protected void onStop() { super.onStop();
/*Code goes here*/
}
/** Called just before the activity is
destroyed. */ @Override
public void onDestroy() { super.onDestroy();
/*Code goes here*/
}
}

```

IX. Practical Related Questions

1. Draw the activity life cycle diagram.

Ans.



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

Ans.

Hierarchy-

1. PROK[project namefolder]
2. app[Folder]
3. Java[Folder]
4. com.example PROK[Folder]
5. mainActivity[.Javafile]

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

Ans.

onStop()	onDestroy()
Called when you are no longer visible to the user. You will next receive either onRestart(), onDestroy(), or nothing, depending on later user activity. Note that this method may never be called, in low memory situations where the system does not have enough memory to keep your activity's process running after its onPause() method is called.	The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called finish() on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the isFinishing() method.

onPause()	onResume()
Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed. The counterpart to onResume(). When activity B is launched in front of activity A, this callback will be invoked on A. B will not be created until A's onPause() returns, so be sure to not do anything lengthy here.	Called when the activity will start interacting with the user. At this point your activity is at the top of the activity stack, with user input going to it. Always followed by onPause().

X. Exercise

1. Write a program to create a HelloWorld Activity using all lifecycles methods to display messages using Log.d.

Ans.

Activity_main.xml:

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

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

MainActivity.java

```
package com.example.practical17;

import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("Activity Lifecycle","onCreate invoked");
    }
    @Override
    protected void onStart()
    {
        super.onStart();
        Log.d("Activity Lifecycle","onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("Activity Lifecycle","onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("Activity Lifecycle","onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("Activity Lifecycle","onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
        Log.d("Activity Lifecycle","onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
}
```

```
        Log.d("Activity Lifecycle","onDestroy invoked");
    }

}
```

Output:

```
2021-05-29 22:09:15.462 5270-5270/com.example.practical17 D/Activity Lifecycle: onPause invoked
2021-05-29 22:09:16.427 5270-5270/com.example.practical17 D/Activity Lifecycle: onStop invoked
2021-05-29 22:09:16.431 5270-5270/com.example.practical17 D/Activity Lifecycle: onDestroy invoked
2021-05-29 22:09:20.523 5270-5270/com.example.practical17 D/Activity Lifecycle: onCreate invoked
2021-05-29 22:09:20.528 5270-5270/com.example.practical17 D/Activity Lifecycle: onStart invoked
2021-05-29 22:09:20.529 5270-5270/com.example.practical17 D/Activity Lifecycle: onResume invoked
2021-05-29 22:10:35.041 5270-5270/com.example.practical17 D/Activity Lifecycle: onPause invoked
2021-05-29 22:10:35.497 5270-5270/com.example.practical17 D/Activity Lifecycle: onStop invoked
2021-05-29 22:10:36.111 5270-5270/com.example.practical17 D/Activity Lifecycle: onRestart invoked
2021-05-29 22:10:36.115 5270-5270/com.example.practical17 D/Activity Lifecycle: onStart invoked
2021-05-29 22:10:36.117 5270-5270/com.example.practical17 D/Activity Lifecycle: onResume invoked
```



Practical No. 18

Develop a program to implement new activity using explicit intent and implicit intent.

35 Siddharth Revankar TYCO

I. Practical Significance

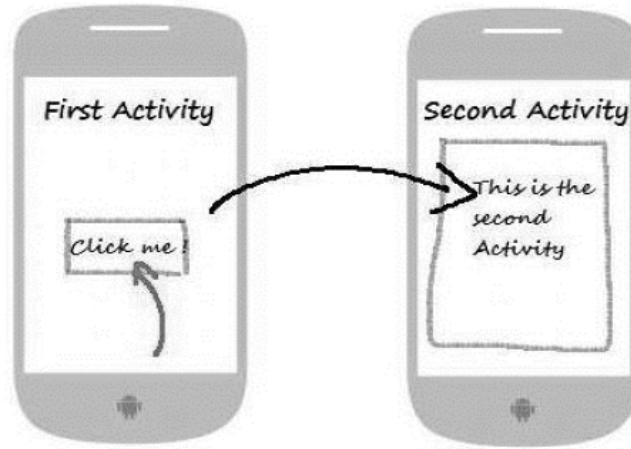
Android Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc. It facilitates communication between different android components.

VII. Minimum Theoretical Background

Android application components can connect to other Android applications. This connection is based on a task description represented by an Intent object. Intents are asynchronous messages which allow application components to request functionality from other Android components. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. For example, an activity can start an external activity for taking a picture.

Mostly Intents are used for:

- a) For Launching an Activity
- b) To start a New Service
- c) For Broadcasting Messages
- d) To Display a list of contacts in List View



Types of intents:

There are two types of intents

- a) Implicit intent
- b) Explicit Intent

Implicit Intent: The implicit intent is the intent where instead of defining the exact components, you define the action that you want to perform for different activities.

Syntax:

```
Intent i = new Intent();
i.setAction(intent.ACTION_SEND);
```

Explicit Intent: An explicit intent is an Intent where you explicitly define the component that needs to be called by the Android System. An explicit intent is one that you can use to launch a specific app component, such as a particular activity or service in your app.

Syntax:

```
Intent i = new Intent(getApplicationContext()).NextActivity.class);
i.putExtra("value1", "This value for Next Activity");
i.putExtra("value2", "This value for Next Activity");
```

IX. Practical Related Questions**1. List different methods used in Intent.****Ans. Different methods used in intent:**

1. putExtra(String s, int i)
2. putExtra(String s, String str)
3. putExtra(String s, boolean b)
4. putExtra(String s, char c)
5. putExtra(String s, float f)
6. putExtra(String s, double d)
7. getStringExtra(String s)
8. getBooleanExtra(boolean b)
9. getIntExtra(int i)

2. Write an intent to display the phone dialer with the given number filled in.**Ans. Intent to display the phone dialer:**

```
Uri u = Uri.parse("tel:+91" + e.getText().toString());
Intent i = new Intent(Intent.ACTION_DIAL, u);
startActivity(i);
```

X. Exercise

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.

Ans.**Activity_main.xml:**

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

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginTop="250dp"
    android:layout_marginEnd="50dp"
    android:ems="10"
    android:hint="Enter url..."
```

```
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/navigateButton"
    android:layout_width="133dp"
    android:layout_height="46dp"
    android:layout_marginStart="100dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="100dp"
    android:backgroundTint="@color/purple_200"
    android:text="Navigate"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
package com.example.practical18;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    private EditText editText;
    private Button navigateButton;

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

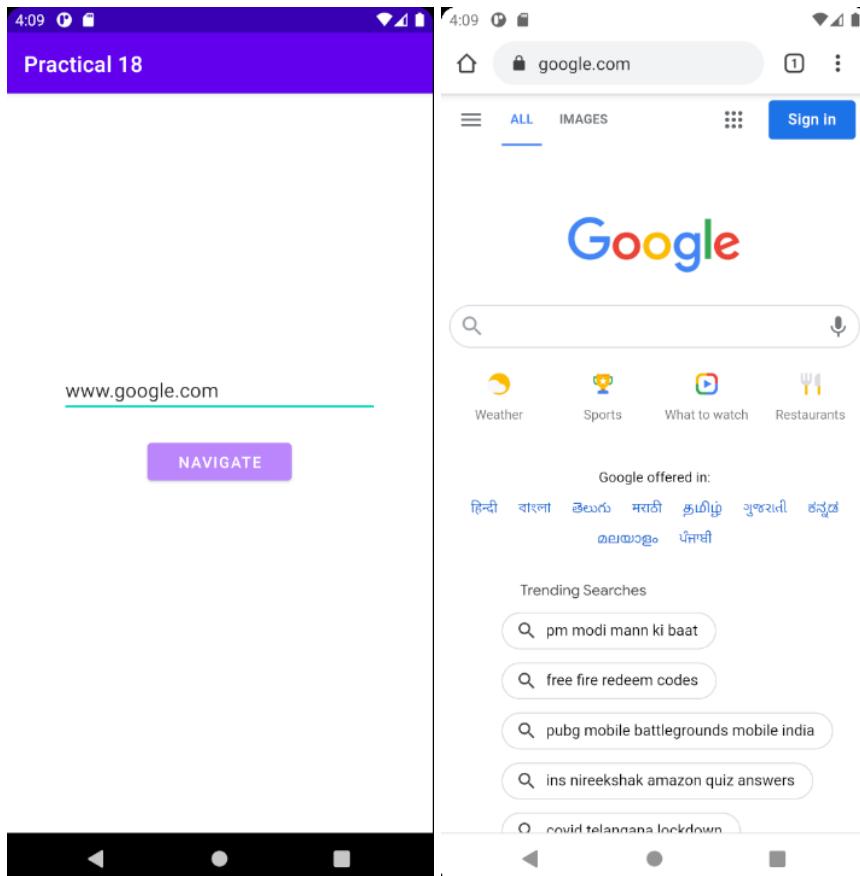
        editText = (EditText) findViewById(R.id.editText);
        navigateButton = (Button) findViewById(R.id.navigateButton);
```

```

navigateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String url = "http://" + editText.getText().toString().trim();
        Intent intent = new Intent();
        intent.setPackage("com.android.chrome");
        intent.setAction(Intent.ACTION_VIEW);
        intent.setData(Uri.parse(url));
        startActivity(intent);
    }
});
}

```

Output:



2. Write a program to create button “Start Dialer”. When u click on this button it should open the phone dialer.

Ans.

Activity_main.xml:

```

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

```

```
        android:layout_height="match_parent"
        tools:context=".MainActivity">

    <EditText
        android:id="@+id/dialerEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="250dp"
        android:layout_marginEnd="50dp"
        android:ems="10"
        android:hint="Enter phone number..."
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/startDialerButton"
        android:layout_width="146dp"
        android:layout_height="46dp"
        android:layout_marginStart="100dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="100dp"
        android:backgroundTint="@color/purple_200"
        android:text="Start Dialer"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dialerEditText" />
```

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

MainActivity.java:

```
package com.example.practical18;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
```

```

private EditText editText;
private Button startDialerButton;

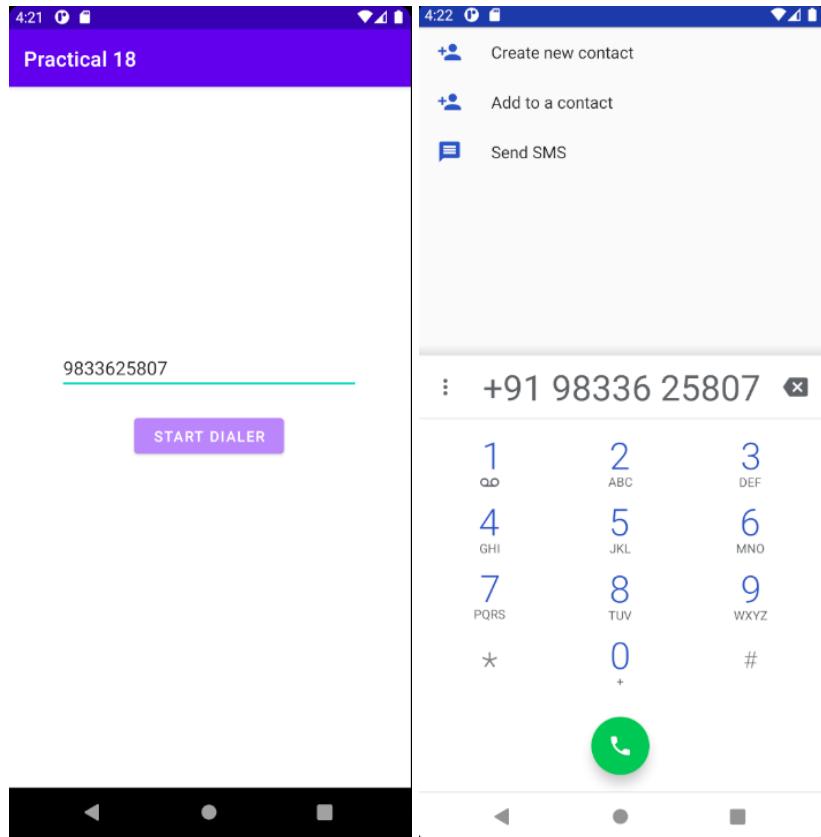
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    editText = (EditText) findViewById(R.id.dialerEditText);
    startDialerButton = (Button) findViewById(R.id.startDialerButton);

    startDialerButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String phoneNumber = editText.getText().toString().trim();
            Uri uri = Uri.parse("tel:+91" + phoneNumber);
            Intent intent = new Intent(Intent.ACTION_DIAL, uri);
            startActivity(intent);
        }
    });
}
}

```

Output:



- 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.**

Ans. Used Intent: Explicit Intent

Activity_main.xml:

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

    <EditText
        android:id="@+id/factInput"
        android:layout_width="143dp"
        android:layout_height="48dp"
        android:layout_marginTop="184dp"
        android:ems="10"
        android:gravity="center"
        android:hint="Enter Number"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/factCalculate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="161dp"
        android:layout_marginLeft="161dp"
        android:layout_marginTop="266dp"
        android:layout_marginEnd="162dp"
        android:layout_marginRight="162dp"
        android:layout_marginBottom="418dp"
        android:backgroundTint="@color/purple_200"
        android:text="Input"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
Activity_main2.xml:
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
        android:id="@+id/factorial"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Factorial"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.301" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
package com.example.practical18;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {

    private EditText factInput;
    private Button factCalculate;

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

        factInput = (EditText) findViewById(R.id.factInput);
```

```
factCalculate = (Button) findViewById(R.id.factCalculate);

factCalculate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String input = factInput.getText().toString().trim();
        Intent intent = new Intent(MainActivity.this, MainActivity2.class);
        intent.putExtra("input", input);
        startActivity(intent);
    }
});
```

MainActivity2.java:

```
package com.example.practical18;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class MainActivity2 extends AppCompatActivity {

    private TextView factorial;

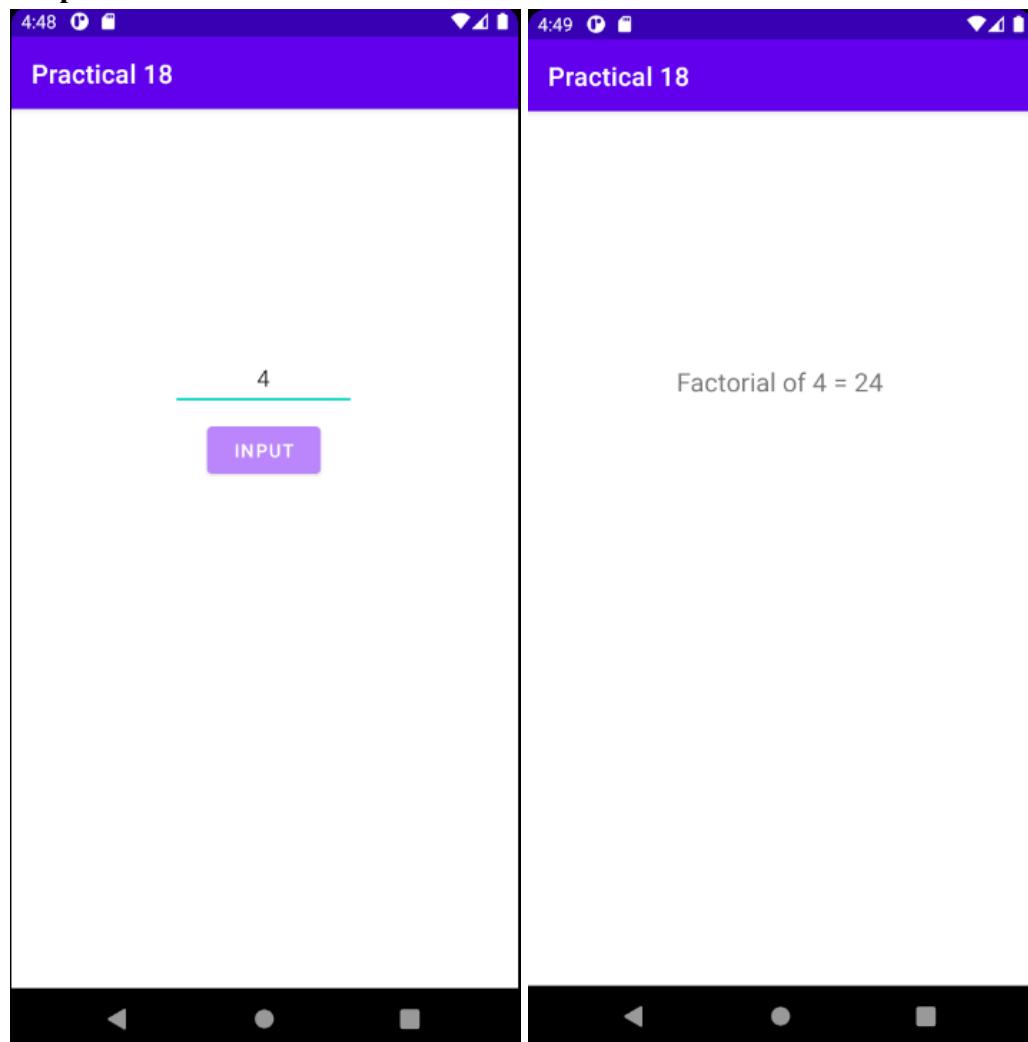
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        String input = getIntent().getStringExtra("input");
        int number = Integer.parseInt(input);

        factorial = (TextView) findViewById(R.id.factorial);

        int i, fact = 1;
        for(i=1; i<=number; i++)
            fact = fact * i;
        factorial.setText("Factorial of " + input + " = " + fact);
    }
}
```

Output:



Practical No. 19

**Develop a program to implement new activity using explicit intent
and implicit intent.**

35 Siddharth Revankar TYCO

I. Practical Significance

Content Provider is one of the pillars of Android. It is a component which

- i)- hides database details (database name, table name, column info. etc.)
- ii)- allows the application to share data among multiple applications.

VII. Minimum Theoretical Background

A content provider component supplies data from one application to other on request. Such requests are handled by the methods of the Content Resolver class. A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network. Content providers let you centralize content in one place and have many different applications access it as needed. A content provider behaves very much like a database where you can query it, edit its content, as well as add or delete content using insert(), update(), delete(), and query() methods. In most cases this data is stored in an SQLite database. A content provider is implemented as a subclass of Content Provider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
public class MyApp extends Content Provider
{
}
```

IX. Practical Related Questions

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

Ans. Here is the list of methods which you need to override in Content Provider class to have your Content Provider working –

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

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

Ans. <prefix>://<authority>/<data_type>/<id>

- prefix:- This is always set to content://
- authority:- This specifies the name of the content provider, for example contacts, browser etc. For third-party content providers, this could be the fully qualified name, such as com.tutorialspoint.statusprovider
- data_type:- This indicates the type of data that this particular provider provides. For example, if you are getting all the contacts from the Contacts content provider, then the data path would be people and URI would look like this content://contacts/people
- id:- This specifies the specific record requested. For example, if you are looking for contact number 5 in the Contacts content provider then URI would look like this content://contacts/people/5.

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

Ans.

- First of all you need to create a Content Provider class that extends the ContentProviderbaseclass.
- Second, you need to define your content provider URI address which will be used to access the content.
- Next you will need to create your own database to keep the content. Usually, Android uses SQLite database and framework needs to override onCreate() method which will use SQLite Open Helper method to create or open the provider's database. When your application is launched, the onCreate() handler of each of its Content Providers is called on the main application thread.
- Next you will have to implement Content Provider queries to perform different database specific operations.
- Finally register your Content Provider in your activity file using <provider> tag.

X. Exercise

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

Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.prac19.MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Content provider"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:paddingTop="50dp"
        android:textSize="30dp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:text="Add Name"
        android:layout_below="@+id/editText3"
        android:layout_alignRight="@+id/textView1"
        android:layout_alignEnd="@+id/textView1"
        android:layout_alignLeft="@+id/textView1"
        android:layout_alignStart="@+id/textView1"
        android:onClick="onClickAddName"/>
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_below="@+id/textView1"
    android:layout_alignRight="@+id/textView1"
    android:layout_alignEnd="@+id/textView1" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:layout_alignTop="@+id/editText"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignStart="@+id/textView1"
    android:layout_alignRight="@+id/textView1"
    android:layout_alignEnd="@+id/textView1"
    android:hint="Name"
    android:textColorHint="@android:color/holo_blue_light" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText3"
    android:layout_below="@+id/editText"
    android:layout_alignLeft="@+id/editText2"
    android:layout_alignStart="@+id/editText2"
    android:layout_alignRight="@+id/editText2"
    android:layout_alignEnd="@+id/editText2"
    android:hint="Grade"
    android:textColorHint="@android:color/holo_blue_bright" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Retrive student"
    android:id="@+id/button"
    android:layout_below="@+id/button2"
    android:layout_alignRight="@+id/editText3"
    android:layout_alignEnd="@+id/editText3"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignStart="@+id/button2"
    android:onClick="onClickRetrieveStudents"/>
```

```
</RelativeLayout>
```

MainActivity.java:

```
package com.example.prac19;
```

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

import android.content.ContentValues;
import android.content.CursorLoader;

import android.database.Cursor;

import android.view.Menu;
import android.view.View;

import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {

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

    public void onClickAddName(View view) {
        // Add a new student record
        ContentValues values = new ContentValues();
        values.put(StudentsProvider.NAME,
                  ((EditText)findViewById(R.id.editText2)).getText().toString());

        values.put(StudentsProvider.GRADE,
                  ((EditText)findViewById(R.id.editText3)).getText().toString());

        Uri uri = getContentResolver().insert(
                  StudentsProvider.CONTENT_URI, values);

        Toast.makeText(getApplicationContext(), "Record inserted successfully!",
                  Toast.LENGTH_LONG).show();
    }

    public void onClickRetrieveStudents(View view) {
        // Retrieve student records
        String URL = "content://com.example.MyApplication.StudentsProvider";

        Uri students = Uri.parse(URL);
        Cursor c = managedQuery(students, null, null, null, "NAME");

        StringBuffer buffer=new StringBuffer();

```

```

        if (c.moveToFirst()) {
            do{
                buffer.append("ID: "+c.getString(c.getColumnIndex(StudentsProvider._ID))+"\n");
                buffer.append("Name: "+c.getString(c.getColumnIndex(StudentsProvider.NAME))+"\n");
                buffer.append("Marks: "+c.getString(c.getColumnIndex(StudentsProvider.GRADE))+"\n\n");
            } while (c.moveToNext());
        }
        Toast.makeText(this,"Student Details:"+buffer.toString(),Toast.LENGTH_SHORT).show();
    }
}

```

StudentsProvider.java

```

package com.example.prac19;

import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;

import android.content.ContentValues;
import android.content.CursorLoader;

import android.database.Cursor;

import android.view.Menu;
import android.view.View;

import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {

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

    public void onClickAddName(View view) {
        // Add a new student record
        ContentValues values = new ContentValues();
        values.put(StudentsProvider.NAME,
                  ((EditText)findViewById(R.id.editText2)).getText().toString());

        values.put(StudentsProvider.GRADE,

```

```

((EditText)findViewById(R.id.editText3)).getText().toString()));

Uri uri = getContentResolver().insert(
    StudentsProvider.CONTENT_URI, values);

Toast.makeText(getApplicationContext(), "Record inserted successfully!",
Toast.LENGTH_LONG).show();
}

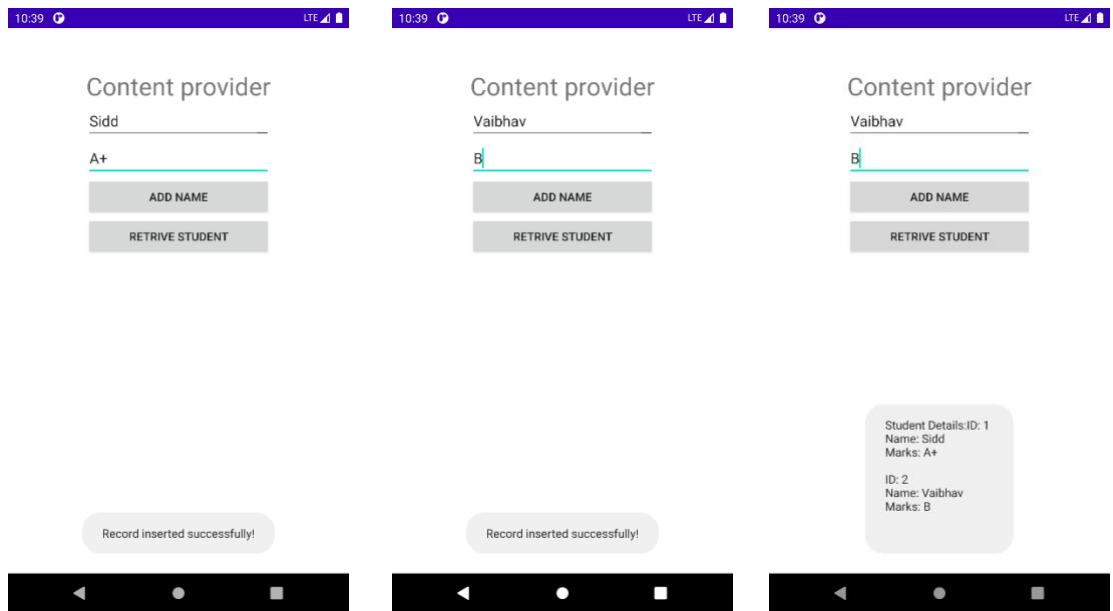
public void onClickRetrieveStudents(View view) {
    // Retrieve student records
    String URL = "content://com.example.MyApplication.StudentsProvider";

    Uri students = Uri.parse(URL);
    Cursor c = managedQuery(students, null, null, null, "NAME");

    StringBuffer buffer=new StringBuffer();
    if (c.moveToFirst()) {
        do{
            buffer.append("ID: "+c.getString(c.getColumnIndex(StudentsProvider._ID))+"\n");
            buffer.append("Name: "+c.getString(c.getColumnIndex(
StudentsProvider.NAME))+"\n");
            buffer.append("Marks: "+c.getString(c.getColumnIndex(
StudentsProvider.GRADE))+"\n\n");
        } while (c.moveToNext());
    }
    Toast.makeText(this,"Student Details:\n\n"+
buffer.toString(),Toast.LENGTH_SHORT).show();
}
}

```

Output:



Practical 20

Aim: Develop a program to implement service.

35 Siddharth Revankar TYCO

Practical Significance:

In android, Service is a component which keep an app running in the background to perform long running operations based on our requirements. For Service, we don't have any user interface and it will run the apps in background like playing the music in background or handle network operations when the user in different app.

Minimum Theoretical Background:

A service is a component which runs in the background without direct interaction with the user. As the service has no user interface, it is not bound to the lifecycle of an activity.

Services are used for repetitive and potentially long running operations, i.e., Internet downloads, checking for new data, data processing, updating content providers. Services run with a higher priority than inactive or invisible activities and therefore it is less likely that the Android system terminates them. Services can also be configured to be restarted if they get terminated by the Android system once sufficient system resources are available again.

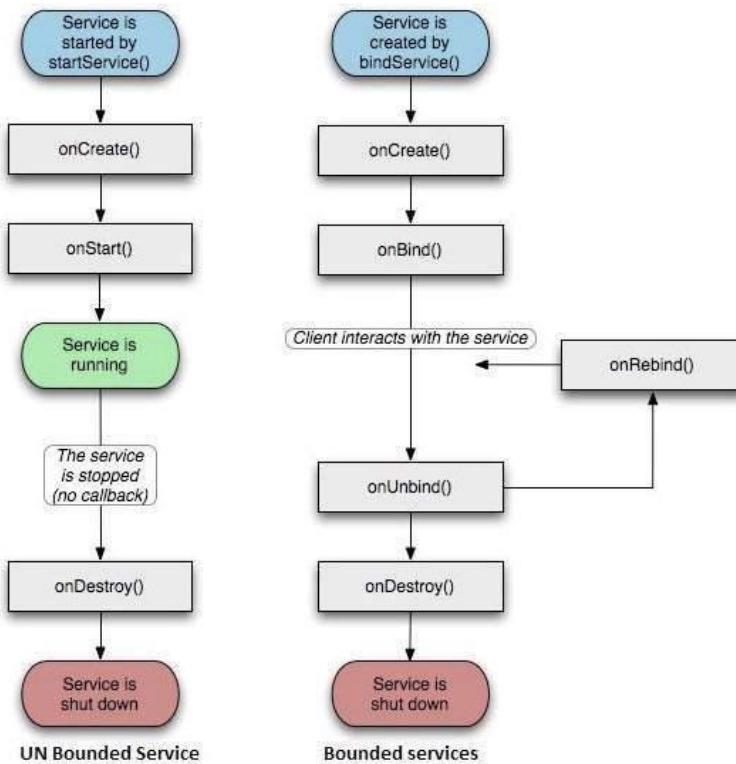
There are the three different types of services:

1. Foreground service
2. Background service
3. Bound service

X Practical Related Questions

1. Draw the lifecycle of service.

Ans.



2. Differentiate between bounded service and unbounded service.

Ans.

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

3. Describe startService() and bindService()methods.

Ans. A service is started when an application component, such as an activity, starts it by calling startService(). Once started, a service can run in the background indefinitely, even if the component that started it is destroyed.

A service is bound when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, get results, and even do so across processes with interprocess communication (IPC).

XI Exercise

1. Write a program to start a Wi-Fi using service.

Ans.

Activity_main.xml

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

```
<Button
    android:id="@+id/enableWifi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="536dp"
    android:text="ON"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<Button
    android:id="@+id/disableWifi"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="460dp"
        android:text="OFF"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent" />

    </androidx.constraintlayout.widget.ConstraintLayout>
MainActivity.java
package com.example.practical20;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private Button enable, disable;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        enable = (Button) findViewById(R.id.enableWifi);
        disable = (Button) findViewById(R.id.disableWifi);

        enable.setOnClickListener(v -> {
            WifiManager wifi = (WifiManager)
                    getApplicationContext().getSystemService(Context.WIFI_SERVICE);
            wifi.setWifiEnabled(true);
            Toast.makeText(MainActivity.this, "Wifi ON!", Toast.LENGTH_SHORT).show();
        });

        disable.setOnClickListener(v -> {
            WifiManager wifi = (WifiManager)
                    getApplicationContext().getSystemService(Context.WIFI_SERVICE);
            wifi.setWifiEnabled(false);
            Toast.makeText(MainActivity.this, "Wifi OFF!", Toast.LENGTH_SHORT).show();
        });
    }
}

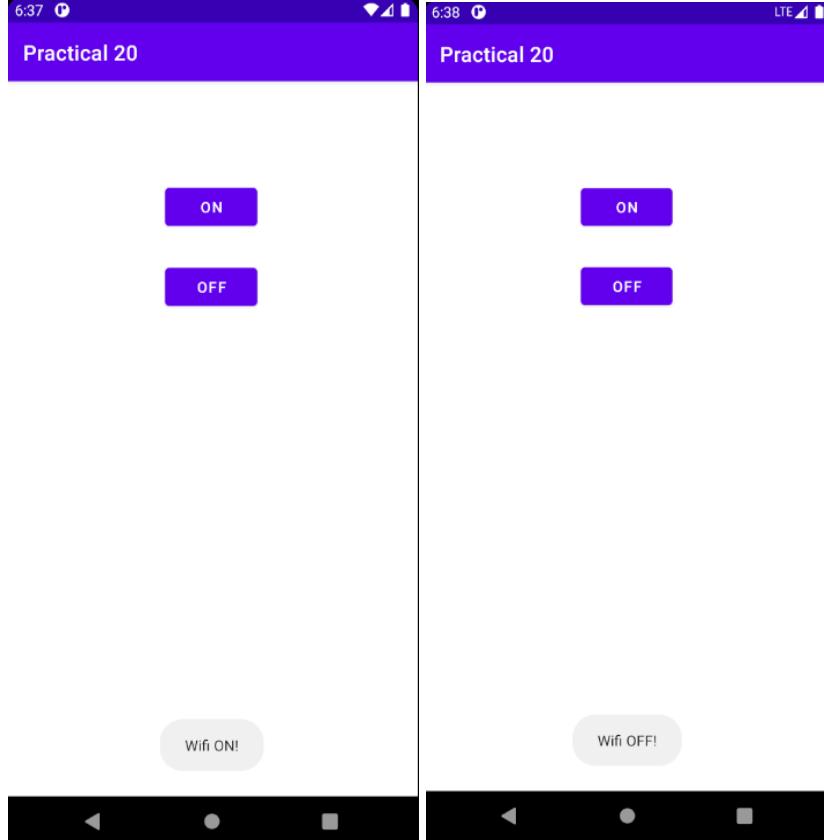
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical20">
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Practical20">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Output:

2. Write a program to display the following output.



Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/button1"
        android:layout_width="380dp"
        android:layout_height="50dp"
        android:text="Start Service"
        app:backgroundTint="#888686"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.516"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.098" />
    <Button
        android:id="@+id/button2"
        android:layout_width="380dp"
        android:layout_height="50dp"
        android:layout_marginTop="40dp"
        android:text="Stop Service"
        app:backgroundTint="#888686"
```

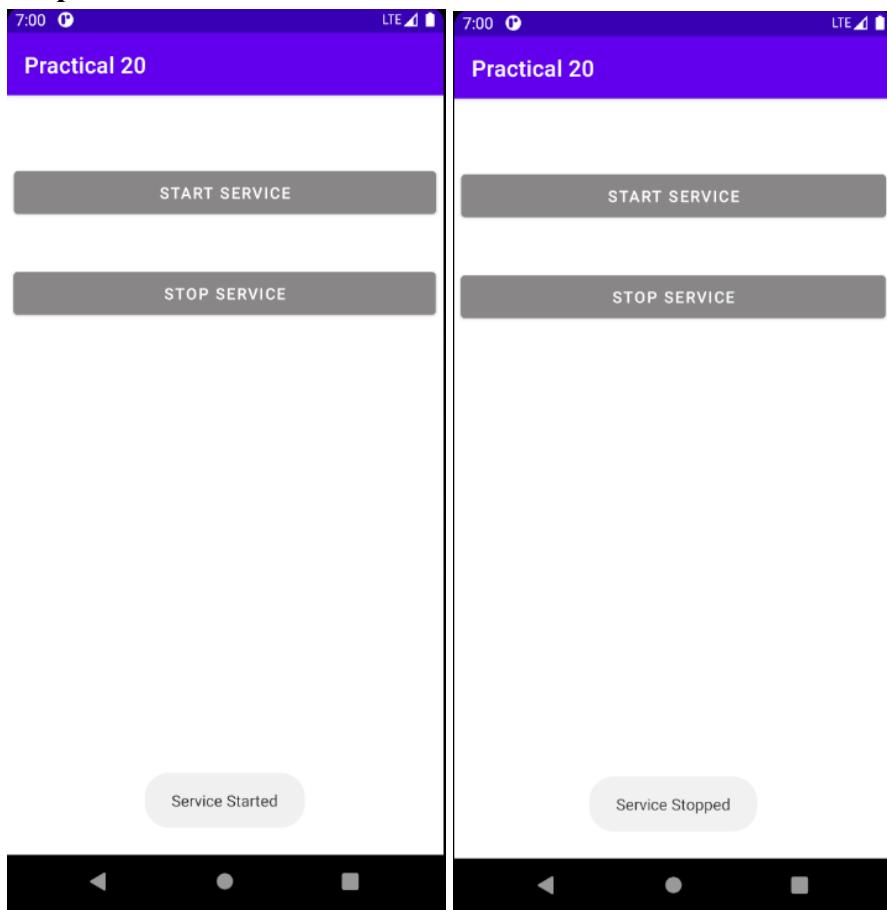
```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.483"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button1" />
</androidx.constraintlayout.widget.ConstraintLayout>
service1.java:
package com.example.practical20;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.widget.Toast;
import androidx.annotation.Nullable;
public class service1 extends Service {
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_SHORT).show();
        return START_STICKY;
    }
    @Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "Service Stopped", Toast.LENGTH_SHORT).show();
    }
}
```

MainActivity.java

```
package com.example.practical20;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    private Button button1, button2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);
```

```
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        startService(v);
    }
});
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        stopService(v);
    }
});
}
public void startService(View view) {
    startService(new Intent(MainActivity.this, service1.class));
}
public void stopService(View view) {
    stopService(new Intent(MainActivity.this, service1.class));
}
}
```

Output:



Practical No.21

Develop a program to implement broadcast receiver

35 Siddharth Revankar TYCO

I. Practical Significance

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

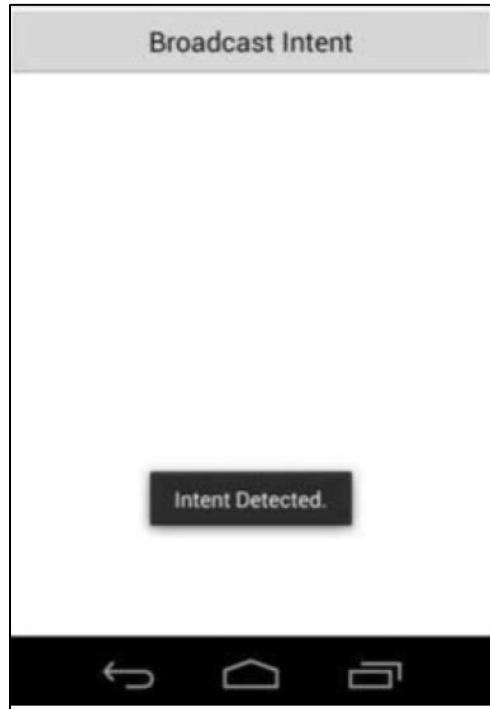
VII. Minimum Theoretical Background

Creating the Broadcast Receiver:

A broadcast receiver is implemented as a subclass of Broadcast Receiver class and overriding the onReceive() method where each message is received as an Intent object parameter.

Registering Broadcast Receiver:

An application listens for specific broadcast intents by registering a broadcast receiver in AndroidManifest.xml file. Consider we are going to register MyReceiver for system generated event ACTION_BOOT_COMPLETED which is fired by the system once the Android system has completed the boot process.



IX. Practical Related Questions

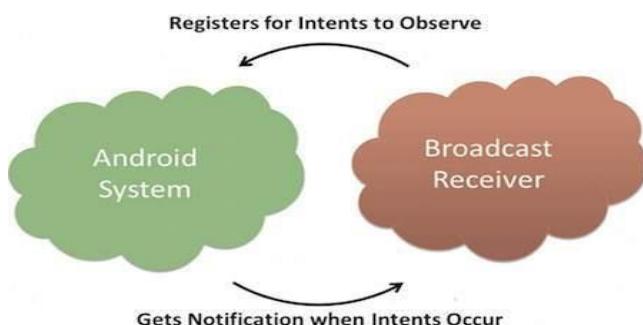
1. Differentiated between Activity Intent and Broadcasting Intent.

Ans.

Activity	Broadcast
Android Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc.	Broadcast intents are Intent objects that are broadcast via a call to the sendBroadcast(),
It is generally used with startActivity() method to invoke activity	It is generally used with sendBroadcast() method

2. Draw Broadcast Receivers Lifecycle.

Ans.



3. List the System Events related to Broadcast Receivers.

Ans.

- i. android.intent.action.BATTERY_CHANGED
- ii. android.intent.action.BATTERY_LOW
- iii. android.intent.action.BATTERY_OKAY
- iv. android.intent.action.BOOT_COMPLETED
- v. android.intent.action.BUG_REPORT
- vi. android.intent.action.CALL
- vii. android.intent.action.CALL_BUTTON
- viii. android.intent.action.DATE_CHANGED
- ix. android.intent.action.REBOOT

X. Exercise

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

Ans.

Activity_main.xml

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

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Practical 21"
        android:textAppearance="@style/TextAppearance.AppCompat.Display2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.prac21">
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Prac21">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".IncommingCallReceiver" android:enabled="true">
            <intent-filter>
                <action android:name="android.intent.action.PHONE_STATE" />
            </intent-filter>
        </receiver>
    
```

```
</application>
```

```
</manifest>
```

MainActivity.java

```
package com.example.prac21;
```

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

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

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

```
}
```

IncommingCallReceiver.java

```
package com.example.prac21;
```

```
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.telephony.TelephonyManager;  
import android.widget.Toast;
```

```
public class IncommingCallReceiver extends BroadcastReceiver{
```

```
    Context context;
```

```
    @Override
```

```
    public void onReceive(Context context, Intent intent){
```

```
        try{
```

```
            String state = intent.getStringExtra(TelephonyManager.EXTRA_STATE);
```

```
            if(state.equals(TelephonyManager.EXTRA_STATE_RINGING)){
```

```
                Toast.makeText(context, "Phone Is Ringing", Toast.LENGTH_LONG).show();
```

```
            }
```

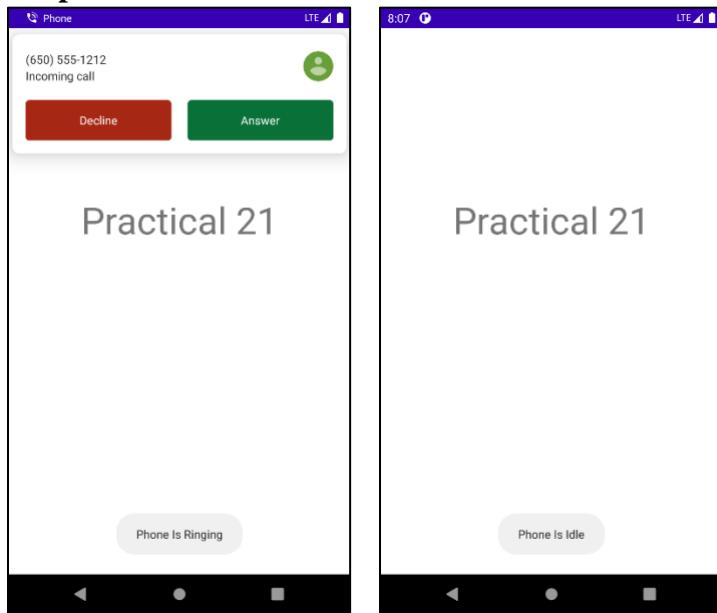
```
if(state.equals(TelephonyManager.EXTRA_STATE_OFFHOOK)){
    Toast.makeText(context, "Call Recieved", Toast.LENGTH_LONG).show();
}

if (state.equals(TelephonyManager.EXTRA_STATE_IDLE)){
    Toast.makeText(context, "Phone Is Idle", Toast.LENGTH_LONG).show();
}

catch(Exception e){e.printStackTrace();}
}

}
```

Output:



Practical No.22

Aim: Develop a program to implement sensors.

35 Siddharth Revankar TYCO

I. Practical Significance

Most of the android devices have built-in sensors that measure motion, orientation, and various environmental condition. Android allows us to get the raw data from these sensors and use it in our application.

VII. Minimum Theoretical Background

The android platform supports three broad categories of sensors. Motion Sensors, Environmental sensors, Position sensors. Some of the sensors are hardware based and some are software-based sensors. Whatever the sensor is, android allows us to get the raw data from these sensors and use it in our application. For this, android provides us with some classes. Android provides Sensor Manager and Sensor classes to use the sensors in our application. In order to use sensors, first thing you need to do is to instantiate the object of SensorManager class.

Example:

```
SensorManager sMgr;
```

```
sMgr = (SensorManager)this.getSystemService(SENSOR_SERVICE);
```

The next thing you need to do is to instantiate the object of Sensor class by calling the getDefaultSensor() method of the SensorManager class. Its syntax is given below:

```
Sensor light;
```

```
light = sMgr.getDefaultSensor(Sensor.TYPE_LIGHT);
```

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, light,SensorManager.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.

IX. Practical Related Questions

1. List the best practices for accessing and using sensors.

Ans.

We can access sensor available on the device and acquire raw sensor data by using the android sensor framework. The sensor framework is a part of the android Hardware package and include the following classes and interface.

- Sensor manager
- Sensor
- SensorEvent
- SensorEventListener

2. Differentiate between Sensor Class and Sensor Manager Class.

Ans.

SensorManager Class	Sensor Class
This is used to get access to various sensors present in the device to use it according to need.	This class is used to create an instance of a specific sensor.
This class is used to find the details of the sensor events.	This interface can be used to trigger or perform some action when there is a change in the sensor values.
Public abstract class sensor manager extends object.	Public final class sensor extends object.

X. Exercise

1. Write a program to changes the background color when device is shuffled.

Ans.

Activity_main.xml

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

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">>

<TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:text="Shake device!"
    android:gravity="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

MainActivity.java

```
package com.example.prac22q1;

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.view.Window;
import android.view.WindowManager;
import android.widget.Toast;

public class MainActivity extends Activity implements SensorEventListener {
    private SensorManager sensorManager;
    private boolean color = false;
    private View view;
    private long lastUpdate;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```

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 onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        getAccelerometer(event);
    }
}

private void getAccelerometer(SensorEvent event) {
    float[] values = event.values;
    // Movement
    float x = values[0];
    float y = values[1];
    float z = values[2];

    float accelerationSquareRoot = (x * x + y * y + z * z)
        / (SensorManager.GRAVITY_EARTH * SensorManager.GRAVITY_EARTH);
    long actualTime = event.timestamp;
    if (accelerationSquareRoot >= 2) //
    {
        if (actualTime - lastUpdate < 200) {
            return;
        }
        lastUpdate = actualTime;
        Toast.makeText(this, "Device was shuffled", Toast.LENGTH_SHORT)
            .show();
        if (color) {
            view.setBackgroundColor(Color.GREEN);
        } else {
            view.setBackgroundColor(Color.RED);
        }
        color = !color;
    }
}

```

```

    }

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {

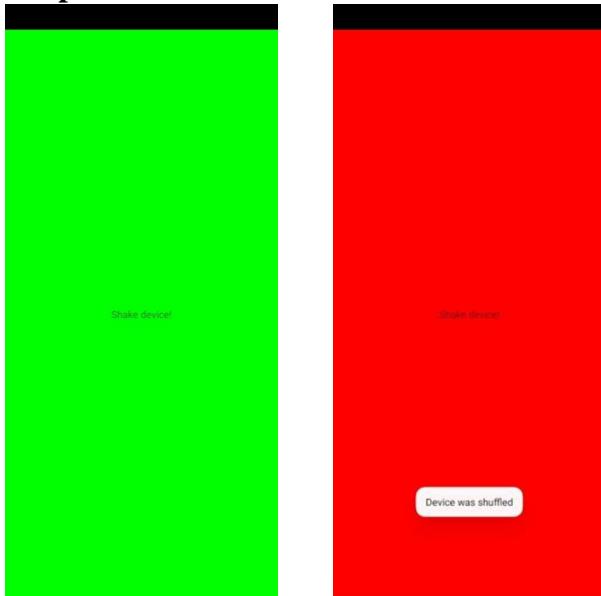
}

@Override
protected void onResume() {
    super.onResume();
    // register this class as a listener for the orientation and
    // accelerometer sensors
    sensorManager.registerListener(this,
        sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        SensorManager.SENSOR_DELAY_NORMAL);
}

@Override
protected void onPause() {
    // unregister listener
    super.onPause();
    sensorManager.unregisterListener(this);
}
}

```

Output:



2. Write a program to display the list of sensors supported by the mobile device.

Ans.

Activity_main.xml

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="80dp"
        android:layout_y="10dp"
        android:text="All Sensors Supports by Android"
        android:gravity="center_horizontal"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="680dp"
        android:layout_x="0dp"
        android:layout_y="50dp"
        app:layout_constraintTop_toTopOf="parent" >

        <!--Text View that shall display the sensor information list-->
        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true" />
    </ScrollView>

</AbsoluteLayout>
```

MainActivity.java

```
package com.example.prac22q2;

import androidx.appcompat.app.AppCompatActivity;

import android.hardware.Sensor;
import android.hardware.SensorManager;
import android.os.Bundle;
```

```

import android.widget.TextView;

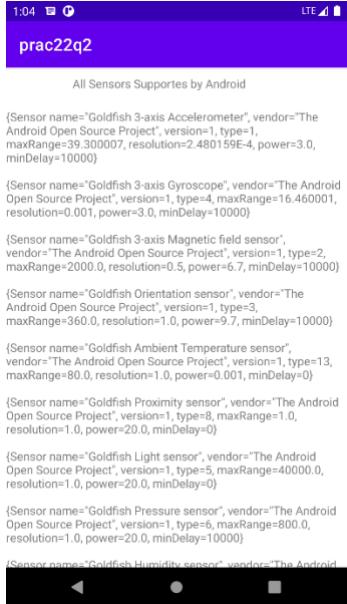
import java.util.List;

public class MainActivity extends AppCompatActivity {
    SensorManager sMgr;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textView=(TextView)findViewById(R.id.textView);
        sMgr = (SensorManager)this.getSystemService(SENSOR_SERVICE);
        List<Sensor> list = sMgr.getSensorList(Sensor.TYPE_ALL);
        for(Sensor sensor: list){
            textView.append(sensor.toString() + "\n\n");
        }
    }
}

```

Output:



Practical No. 23

Aim: Develop a program to build Camera.

35 Siddharth Revankar TYCO

I. Practical Significance

The Android framework includes support for various cameras and camera features available on devices, allowing you to capture pictures and videos in your application.

VII. Minimum Theoretical Background

Camera can be used in your application in following ways.

1. Using existing android camera application in our application
2. Directly using Camera API provided by android in our application

You will use MediaStore.ACTION_IMAGE_CAPTURE to launch an existing camera application installed on your phone. Its syntax is given below:

Intent intent = new

Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);

We will be using the camera API to integrate the camera in our application. First you will need to initialize the camera object using the static method provided by the api called Camera. Open. Its syntax is:

Camera object = null;

object = Camera.open();

Methods:

1. startActivityForResult(Intent intent, int requestCode, Bundle options) It starts an activity, but can take extra bundle of options with it.
2. startActivityFromChild(Activity child, Intent intent, int requestCode) It launches the activity when your activity is child of any otheractivity.
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 currentlyinside.
5. startActivityFromFragment(Fragment fragment, Intent intent, int requestCode, Bundle options) It not only launches the activity from the fragment, but can take extra values withit.

IX. Practical Related Questions

1. List all the methods related to camera class.

Ans.

Methods:

1. addCallbackBuffer(byte[] callbackBuffer)
2. autoFocus(Camera.AutoFocusCallback cb)
3. takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg)
4. takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback postview, Camera.PictureCallback jpeg)
5. setFaceDetectionListener(Camera.FaceDetectionListener listener)
6. setZoomChangeListener(Camera.OnZoomChangeListener listener)
7. getCameraInfo(int cameraId, Camera.CameraInfo cameraInfo)
8. startSmoothZoom(int value)

2. Explain the method that is used to detect the face.

Ans. Method: public FaceDetector (int width, int height, int maxFaces)

Creates a FaceDetector, configured with the size of the images to be analysed and the maximum number of faces that can be detected. These parameters cannot be changed once the object is constructed. Note that the width of the image must be even.

X. Exercise

1. Write a program to capture an image and display it using image view.

Ans.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout 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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.practical23.MainActivity"
    android:orientation="vertical"
    android:background="#FFF9C4">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="fill_parent"
        android:layout_height="300dp"
        android:layout_centerHorizontal="true"
        android:layout_x="0dp"
        android:layout_y="63dp" />

    <Button
        android:id="@+id/button"
        android:layout_width="164dp"
        android:layout_height="wrap_content"
        android:layout_x="142dp"
        android:layout_y="405dp"
        android:text="Capture" />
</AbsoluteLayout>
```

MainActivity.java:

```
package com.example.practical23;
```

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

```
import android.os.Bundle;
import android.Manifest;
import android.content.Intent;
```

```
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button button ;
    ImageView imageView ;
    Intent intent ;
    public static final int RequestPermissionCode = 1 ;

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

        button = (Button)findViewById(R.id.button);
        imageView = (ImageView)findViewById(R.id.imageView);

        EnableRuntimePermission();

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                intent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);

                startActivityForResult(intent, 7);

            }
        });
    }

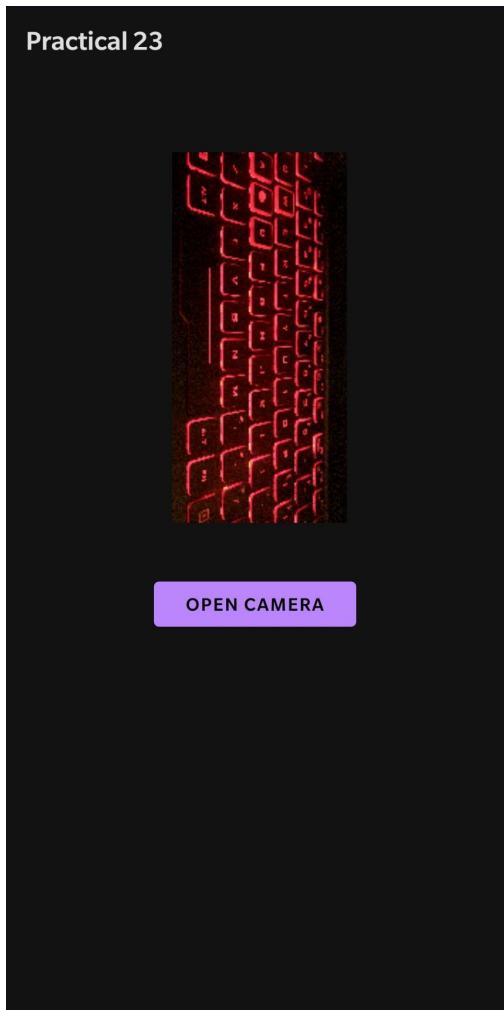
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 7 && resultCode == RESULT_OK) {

            Bitmap bitmap = (Bitmap) data.getExtras().get("data");

            imageView.setImageBitmap(bitmap);
        }
    }
}
```

```
public void EnableRuntimePermission() {
    if (ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,
        Manifest.permission.CAMERA))
    {
        Toast.makeText(MainActivity.this, "CAMERA permission allows us to Access
        CAMERA app", Toast.LENGTH_LONG).show();
    } else {
        ActivityCompat.requestPermissions(MainActivity.this, new String[] {
            Manifest.permission.CAMERA}, RequestPermissionCode);
    }
}

@Override
public void onRequestPermissionsResult(int RC, String per[], int[] PResult) {
    super.onRequestPermissionsResult(RC, per, PResult);
    switch (RC) {
        case RequestPermissionCode:
            if (PResult.length > 0 && PResult[0] ==
                PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(MainActivity.this, "Permission Granted, Now your application
                can access CAMERA.", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(MainActivity.this, "Permission Canceled, Now your application
                cannot access CAMERA.", Toast.LENGTH_LONG).show();
            }
            break;
    }
}
```

Output:**2. Write a program to record a video using various camera methods.**

Ans.

Activity_main.xml:

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

    <LinearLayout
        android:id="@+id/camera_preview"
        android:layout_width="295dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical" />

    <AbsoluteLayout
        android:id="@+id/buttonsLayout"
```

```

        android:layout_width="136dp"
        android:layout_height="266dp"
        android:layout_gravity="center"
        android:orientation="horizontal">

    <Button
        android:id="@+id/button_ChangeCamera"
        style="@style/Widget.AppCompat.Button.Borderless.Colored"
        android:layout_width="wrap_content"
        android:layout_height="45dp"
        android:layout_marginTop="30dp"
        android:layout_x="5dp"
        android:layout_y="50dp"
        android:background="#03A9F4"
        android:text="Switch Camera" />

    <Button
        android:id="@+id/button_capture"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="117dp"
        android:layout_height="58dp"
        android:layout_marginTop="100dp"
        android:layout_x="2dp"
        android:layout_y="128dp"
        android:background="#2196F3"
        android:text="Start/Stop" />
    </AbsoluteLayout>
</LinearLayout>

```

MainActivity.java:

```

package com.example.practical23;
import java.io.IOException;

import android.app.Activity;
import android.content.Context;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.hardware.Camera.CameraInfo;
import android.media.CamcorderProfile;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.LinearLayout;

```

```
import android.widget.Toast;

public class MainActivity 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;
    }
}
```

```

        }
    }
    return cameraId;
}

public void onResume() {
    super.onResume();
    if (!hasCamera(mContext)) {
        Toast toast = Toast.makeText(mContext, "Sorry, your phone does not have a
camera!", Toast.LENGTH_LONG);
        toast.show();
        finish();
    }
    if (mCamera == null) {

        if (findFrontFacingCamera() < 0) {
            Toast.makeText(this, "No front facing camera found.",
Toast.LENGTH_LONG).show();
            switchCamera.setVisibility(View.GONE);
        }
        mCamera = Camera.open(findBackFacingCamera());
        mPreview.refreshCamera(mCamera);
    }
}

public void initialize() {
    cameraPreview = (LinearLayout) findViewById(R.id.camera_preview);

    mPreview = new CameraPreview(mContext, mCamera);
    cameraPreview.addView(mPreview);

    capture = (Button) findViewById(R.id.button_capture);
    capture.setOnClickListener(captrueListener);

    switchCamera = (Button) findViewById(R.id.button_ChangeCamera);
    switchCamera.setOnClickListener(switchCameraListener);
}

OnClickListener switchCameraListener = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!recording) {
            int camerasNumber = Camera.getNumberOfCameras();
            if (camerasNumber > 1) {

```

```
        releaseCamera();
        chooseCamera();
    } else {
        Toast toast = Toast.makeText(mContext, "Sorry, your phone has only one
camera!", Toast.LENGTH_LONG);
        toast.show();
    }
}
};

public void chooseCamera() {

if (cameraFront) {
    int cameraId = findBackFacingCamera();
    if (cameraId >= 0) {

        mCamera = Camera.open(cameraId);
        // mPicture = getPictureCallback();
        mPreview.refreshCamera(mCamera);
    }
} else {
    int cameraId = findFrontFacingCamera();
    if (cameraId >= 0) {

        mCamera = Camera.open(cameraId);
        // mPicture = getPictureCallback();
        mPreview.refreshCamera(mCamera);
    }
}
}

@Override
protected void onPause() {
    super.onPause();
    releaseCamera();
}

private boolean hasCamera(Context context) {
    if
(context.getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA))
{
    return true;
} else {
    return false;
}
}
```

```

}

boolean recording = false;
OnClickListener captureListener = new OnClickListener() {
    @Override
    public void onClick(View v) {
        if (recording) {
            mediaRecorder.stop();
            releaseMediaRecorder();
            Toast.makeText(MainActivity.this, "Video captured!",
Toast.LENGTH_LONG).show();
            recording = false;
        } else {
            if (!prepareMediaRecorder()) {
                Toast.makeText(MainActivity.this, "Fail in prepareMediaRecorder()!\n - Ended",
-", Toast.LENGTH_LONG).show();
                finish();
            }
        }
    }

    runOnUiThread(new Runnable() {
        public void run() {

            try {
                mediaRecorder.start();
            } catch (final Exception ex) {
                // Log.i("---","Exception in thread");
            }
        }
    });

    recording = true;
}
};

private void releaseMediaRecorder() {
    if (mediaRecorder != null) {
        mediaRecorder.reset();
        mediaRecorder.release();
        mediaRecorder = null;
        mCamera.lock();
    }
}

private boolean prepareMediaRecorder() {

```

```
mediaRecorder = new MediaRecorder();

mCamera.unlock();
mediaRecorder.setCamera(mCamera);

mediaRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);
mediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);

mediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_720P));

mediaRecorder.setOutputFile("/sdcard/myvideo.mp4");
mediaRecorder.setMaxDuration(600000);
mediaRecorder.setMaxFileSize(50000000);

try {
    mediaRecorder.prepare();
} catch (IllegalStateException e) {
    releaseMediaRecorder();
    return false;
} catch (IOException e) {
    releaseMediaRecorder();
    return false;
}
return true;

}
```

```
private void releaseCamera() {
```

```
    if (mCamera != null) {
        mCamera.release();
        mCamera = null;
    }
}
```

CameraPreview.java:

```
package com.example.practical23;

import java.io.IOException;

import android.content.Context;
import android.hardware.Camera;
import android.util.Log;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
```

```
public class CameraPreview extends SurfaceView implements SurfaceHolder.Callback {  
    private SurfaceHolder mHolder;  
    private Camera mCamera;  
  
    public CameraPreview(Context context, Camera camera) {  
        super(context);  
        mCamera = camera;  
        mHolder = getHolder();  
        mHolder.addCallback(this);  
        mHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    }  
  
    public void surfaceCreated(SurfaceHolder holder) {  
        try {  
            if (mCamera == null) {  
                mCamera.setPreviewDisplay(holder);  
                mCamera.startPreview();  
            }  
        } catch (IOException e) {  
            Log.d(VIEW_LOG_TAG, "Error setting camera preview: " + e.getMessage());  
        }  
    }  
  
    public void refreshCamera(Camera camera) {  
        if (mHolder.getSurface() == null) {  
            return;  
        }  
        try {  
            mCamera.stopPreview();  
        } catch (Exception e) {  
            }  
        setCamera(camera);  
        try {  
            mCamera.setPreviewDisplay(mHolder);  
            mCamera.startPreview();  
        } catch (Exception e) {  
            Log.d(VIEW_LOG_TAG, "Error starting camera preview: " + e.getMessage());  
        }  
    }  
  
    public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {  
        refreshCamera(mCamera);  
    }  
  
    public void setCamera(Camera camera) {  
        mCamera = camera;
```

```
    }

    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {

    }
}
```

AndroidManifest.xml:

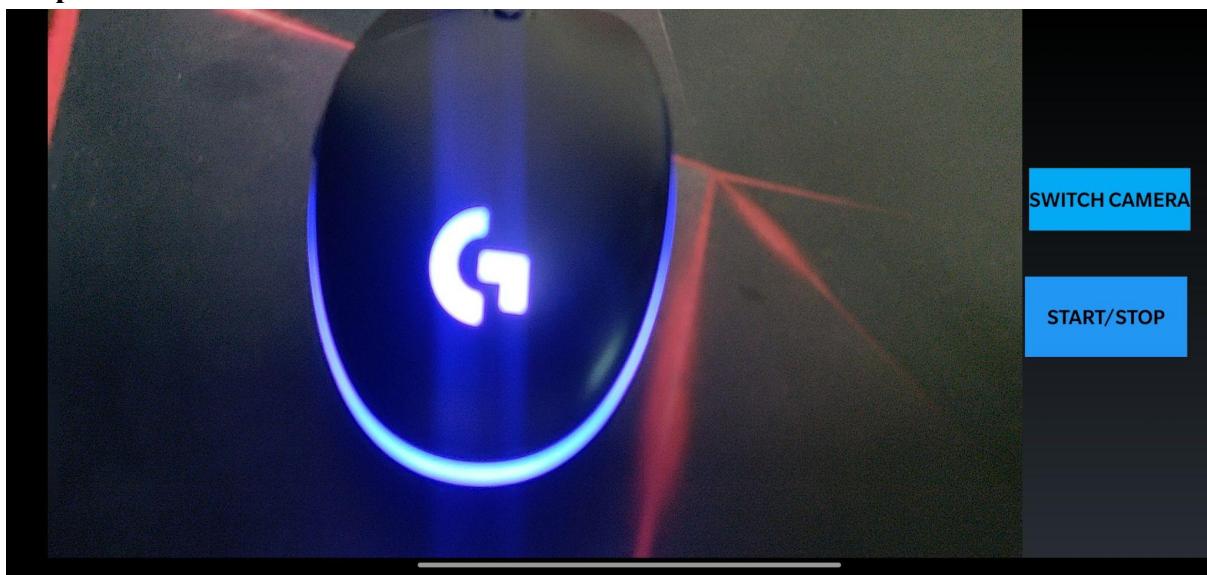
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical23"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />
    <uses-feature
        android:name="android.hardware.camera.front"
        android:required="false" />

    <application
        android:allowBackup="true"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >
        <activity
            android:name="com.example.practical23.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

Output:



Practical No.24
Develop a program for providing Bluetooth connectivity
35 Siddharth Revankar TYCO

I. Practical Significance

Bluetooth is a way to send or receive data between two different devices. Android platform includes support for the Bluetooth framework that allows a device to wirelessly exchange data with other Bluetooth devices.

VII. Minimum Theoretical Background

Android provides Bluetooth API to perform these different operations.

1. Scan for other Bluetooth devices
2. Get a list of paired devices.
3. Connect to other devices through service discovery.

Android provides Bluetooth Adapter class to communicate with Bluetooth. Create an object of this calling by calling the static method getDefaultAdapter(). Its syntax is given below.

```
private BluetoothAdapter BA;  
BA = BluetoothAdapter.getDefaultAdapter();
```

In order to enable the Bluetooth of your device, call the intent with the following Bluetooth constant ACTION_REQUEST_ENABLE. Its syntax is.

```
Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
```

```
startActivityForResult(turnOn, 0);
```

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();
```

IX. Practical Related Questions

1. Name the methods which are used to enable and disable Bluetooth adapter.

Ans. Methods which are used to enable and disable Bluetooth adapter:

- enable()
- disable()

2. Explain the purpose of ACTION_REQUEST_DISCOVERABLE Constant

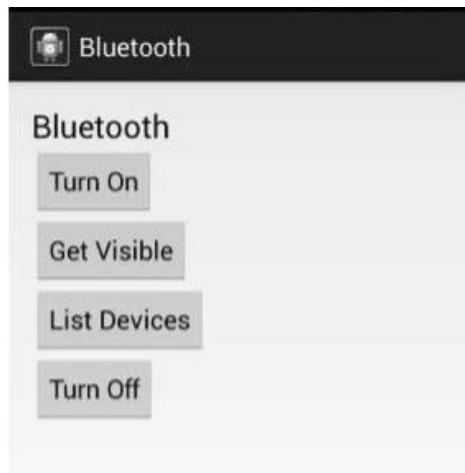
Ans. Activity Action: Show a system activity that requests discoverable mode. This activity will also request the user to turn on Bluetooth if it is not currently enabled. Discoverable mode is equivalent to SCAN_MODE_CONNECTABLE_DISCOVERABLE. It allows remote devices to see this Bluetooth adapter when they perform a discovery.

3. List the uses of setName(String name)method.

Ans. The setName() method provided by the java.lang.Thread class is used to change the name of the thread.

X. Exercise

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



Ans.

Activity_main.xml

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

```
<Button
```

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:clickable="true"
    android:onClick="on"
```

```
        android:text="Turn On"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.027"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.117" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/button"
    android:layout_centerHorizontal="true"
    android:onClick="visible"
    android:text="Get visible"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.025"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.254" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="list"
    android:text="List devices"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.03"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.377" />
```

```
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
```

```
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:onClick="off"
        android:text="turn off"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.052"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />
```

```
<ListView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/listView"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/button"
    android:layout_alignStart="@+id/button"
    android:layout_below="@+id/button"
    tools:ignore="MissingConstraints,UnknownId" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="11dp"
    android:layout_marginLeft="11dp"
    android:layout_marginTop="43dp"
    android:layout_marginEnd="305dp"
    android:layout_marginRight="305dp"
    android:layout_marginBottom="7dp"
    android:text="Bluetooth"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toTopOf="@+id/button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

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

MainActivity.java

```
package com.example.practical24;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ArrayAdapter;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.Set;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {
    Button b1,b2,b3,b4;
    private BluetoothAdapter BA;
    private Set<BluetoothDevice>pairedDevices;
    ListView lv;
    @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.button2);
        b3=(Button)findViewById(R.id.button3);
        b4=(Button)findViewById(R.id.button4);

        BA = BluetoothAdapter.getDefaultAdapter();
        lv = (ListView)findViewById(R.id.listView);
    }
    public void on(View v){
        if (!BA.isEnabled()) {
            Intent turnOn = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(turnOn, 0);
            Toast.makeText(getApplicationContext(), "Turned on",Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(getApplicationContext(), "Already on",
            Toast.LENGTH_LONG).show();
        }
    }
}
```

```

public void off(View v){
    BA.disable();
    Toast.makeText(getApplicationContext(), "Turned off" ,Toast.LENGTH_LONG).show();
}

public void visible(View v){
    Intent getVisible = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
    startActivityForResult(getVisible, 0);
}

public void list(View v){
    pairedDevices = BA.getBondedDevices();

    ArrayList list = new ArrayList();

    for(BluetoothDevice bt : pairedDevices) list.add(bt.getName());
    Toast.makeText(getApplicationContext(), "Showing Paired
Devices",Toast.LENGTH_SHORT).show();

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

    lv.setAdapter(adapter);
}
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical24">
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

```

```

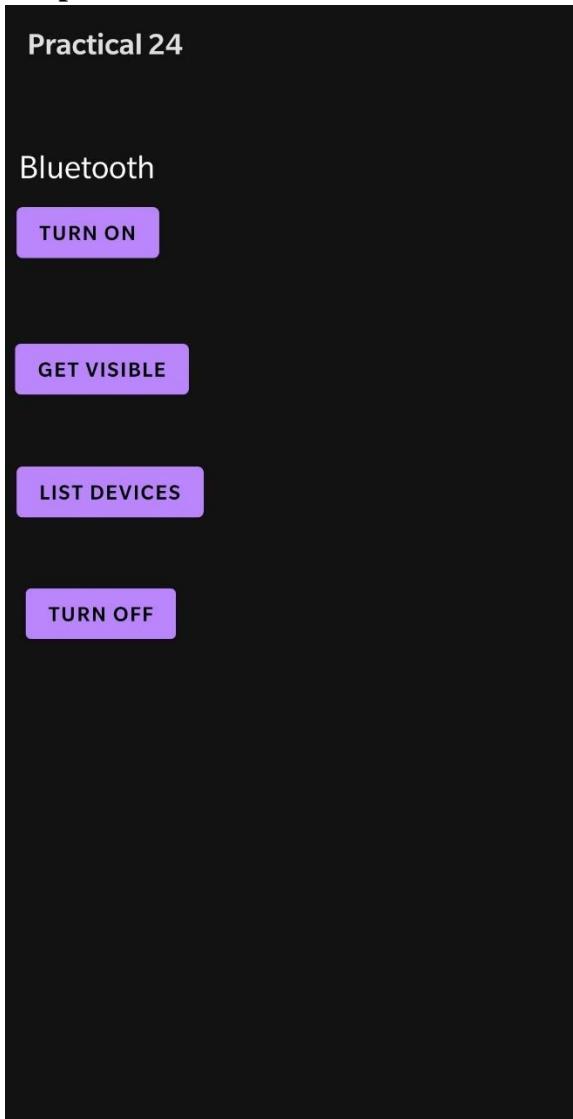
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
```

```
    android:supportsRtl="true"
    android:theme="@style/Theme.Practical24">
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Output:



Practical No.25
Develop a program for animation.
35 Siddharth Revankar TYCO

I. Practical Significance

Animation is the process of creating motion and shape change. Tween Animation takes some parameters such as start value, end value, size, time duration, rotation angle etc., and perform the required animation on that object. It can be applied to any type of object.

VII. Minimum Theoretical Background

In order to perform animation in android, we are going to call a static function loadAnimation() of the class AnimationUtils. We are going to receive the result in an instance of Animation Object. Its syntax is as follows:

```
Animation animation =  
    AnimationUtils.loadAnimation(getApplicationContext(),  
        R.anim.myanimation);
```

In order to apply this animation to an object, we will just call the startAnimation() method of the object. Its syntax is:

```
ImageView image1 = (ImageView) findViewById(R.id.imageView1);  
image.startAnimation(animation);
```

Methods:

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

IX. Practical Related Questions

1. Write the steps to perform Tween Animation

Ans. Steps:

- Step 1: Open or Start an Android Project.
- Step 2: Create the Sky Shape Drawable.
- Step 3: Create the Sun Shape Drawable.
- Step 4: Create the Grass Shape Drawable.
- Step 5: Include the Drawables in the Layout.
- Step 6: Define the Sun Rise Animation.
- Step 7: Apply the Animation.
- Step 8: Add a Clock Animation.

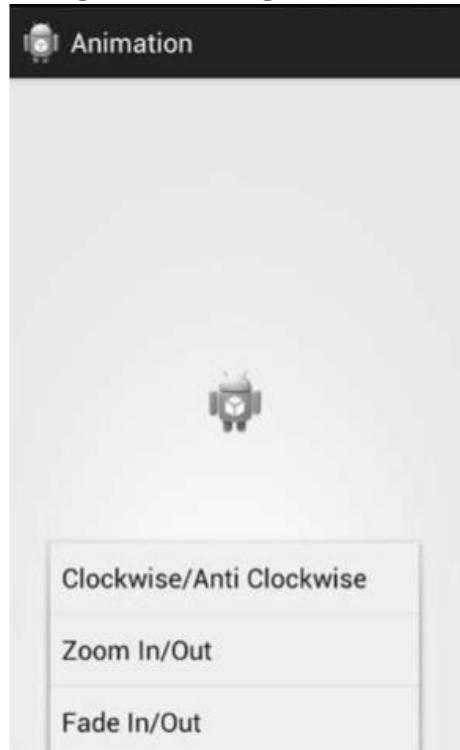
2. Explain the use of from XScale and from YScale method in detail

Ans. 1. fromX: Horizontal scaling factor to apply at the start of the animation

2. fromY: Vertical scaling factor to apply at the start of the animation

X. Exercise

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



Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.practical25.MainActivity">
```

<ImageView

```
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="183dp"
```

```
    android:layout_marginLeft="183dp"
    android:layout_marginTop="288dp"
    android:layout_marginEnd="185dp"
    android:layout_marginRight="185dp"
    android:contentDescription="@string/todo"
    android:src="@drawable/abc"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.517"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="200dp"
    android:onClick="clockwise"
    android:text="@string/zoom"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="33dp"
    android:layout_marginLeft="33dp"
    android:layout_marginTop="200dp"
    android:layout_marginEnd="42dp"
    android:layout_marginRight="42dp"
    android:onClick="zoom"
    android:text="@string/clockwise"
    app:layout_constraintEnd_toStartOf="@+id/button3"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toEndOf="@+id/button"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
<Button
    android:id="@+id/button3"
```

```
    android:layout_width="89dp"
    android:layout_height="47dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:onClick="fade"
    android:text="@string/fade"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button2"
    app:layout_constraintTop_toTopOf="@+id/button2" />
```

<Button

```
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="12dp"
    android:onClick="blink"
    android:text="@string/blink"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />
```

<Button

```
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:onClick="move"
    android:text="@string/move"
    app:layout_constraintEnd_toEndOf="@+id/button2"
    app:layout_constraintHorizontal_bias="0.625"
    app:layout_constraintLeft_toLeftOf="@+id/button2"
    app:layout_constraintRight_toRightOf="@+id/button2"
    app:layout_constraintStart_toStartOf="@+id/button2"
    app:layout_constraintTop_toBottomOf="@+id/button2" />
```

<Button

```
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="54dp"
```

```
        android:layout_marginLeft="54dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="19dp"
        android:layout_marginRight="19dp"
        android:onClick="slide"
        android:text="@string/slide"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/button5"
    app:layout_constraintTop_toBottomOf="@+id/button3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.practical25;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void clockwise(View view){
        ImageView image = findViewById(R.id.imageView);
        Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),
            R.anim.myanimation);
        image.startAnimation(animation);
    }
    public void zoom(View view){
        ImageView image = findViewById(R.id.imageView);
        Animation animation1 = AnimationUtils.loadAnimation(getApplicationContext(),
            R.anim.clockwise);
        image.startAnimation(animation1);
    }
    public void fade(View view){
        ImageView image = findViewById(R.id.imageView);
        Animation animation1 = AnimationUtils.loadAnimation(getApplicationContext(),
            R.anim.fade);
    }
}
```

```

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

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical25">
    <application
        android:allowBackup="true"
        android:icon="@drawable/abc"
        android:label="@string/app_name">
        <activity
            android:name="com.example.practical25.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical25">
    <application>
        android:allowBackup="true"
        android:icon="@drawable/abc"
        android:label="@string/app_name">
            <activity>
                android:name="com.example.practical25.MainActivity"
                android:label="@string/app_name" >
                    <intent-filter>
                        <action android:name="android.intent.action.MAIN" />
                        <category android:name="android.intent.category.LAUNCHER" />
                    </intent-filter>
            </activity>
        </application>
    </manifest>
```

Clockwise.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>
    <rotate
        android:startOffset="5000"
        android:fromDegrees="360"
        android:toDegrees="0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="5000" >
    </rotate>
</set>
```

Fade.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator" >
    <alpha
        android:fromAlpha="0"
        android:toAlpha="1"
```

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

Blink.xml:

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

Move.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">
    <translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
        android:duration="800" />
</set>
```

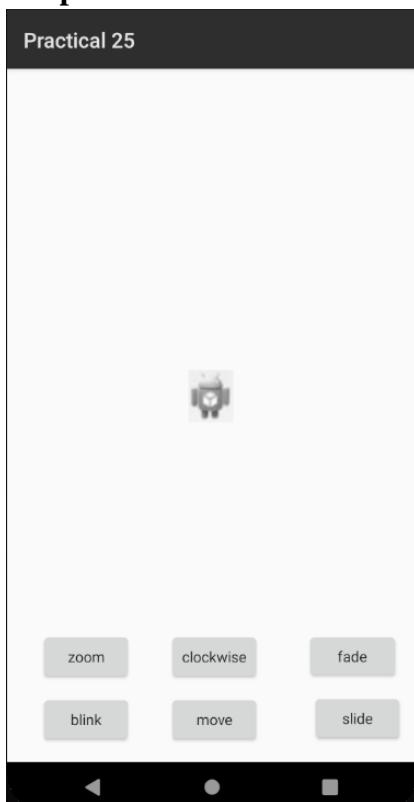
Slide.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```

Myanimation.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <scale
        android:fromXScale="0.5"
        android:toXScale="3.0"
        android:fromYScale="0.5"
        android:toYScale="3.0"
        android:duration="5000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>
    <scale
        android:startOffset="5000"
        android:fromXScale="3.0"
        android:toXScale="0.5"
        android:fromYScale="3.0"
        android:toYScale="0.5"
        android:duration="5000"
        android:pivotX="50%"
        android:pivotY="50%" >
    </scale>
</set>
```

Output:



Practical No. 26
Perform Async task using SQLite.
35 Siddharth Revankar TYCO

I. Practical Significance

Android AsyncTask is an abstract class provided by Android which gives us the liberty to perform heavy tasks in the background and keep the UI thread light thus making the application more responsive.

VII. Minimum Theoretical Background

Android application runs on a single thread when launched. Due to this single thread model tasks that take longer time to fetch the response can make the application nonresponsive. To avoid this, we use android AsyncTask to perform the heavy tasks in background on a dedicated thread and passing the results back to the UI thread. Hence use of AsyncTask in android application keeps the UI thread responsive at all times.

IX. Practical Related Questions

1. List the basic methods used in an android AsyncTask class.

Ans. Basic methods used in an android AsyncTask class

- doInBackground()
- onPreExecute()
- onPostExecute()
- onProgressUpdate()

2. Differentiate between AsyncTask and Services.

Ans.

	AsyncTask	Service
Definition	AsyncTask enables proper and easy use of the UI thread.	A Service is an application component that can perform long-running operations in the background.
When to use?	<ul style="list-style-type: none">• Small task having to communicate with main thread• For tasks in parallel use multiple instances OR Executor• Disk-bound tasks that might take more than a few milliseconds	<ul style="list-style-type: none">• Task with no UI, but shouldn't be too long. Use threads within service for long tasks.• Long task in general.
Trigger	Call to method execute()	Call to method onStartService()
Triggered from	Main Thread	Any Thread
Runs on	Worker thread. However, Main thread methods may be invoked in between to publish progress.	Main thread of its hosting process.

Limitations / Drawbacks	<ul style="list-style-type: none"> • One instance can only be executed once • Must be created and executed from the Main thread 	May block main thread
--------------------------------	---	-----------------------

3. Name the method used, if a process takes a long time to do its work?

Ans. Services that have been running for a long time (such as 30 minutes or more) may be demoted in importance to allow their process to drop to the cached list described next. This helps avoid situations where long running services that use excessive resources (for example, by leaking memory) prevent the system from delivering a good user experience.

X. Exercise

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

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="50dp"
        android:layout_y="20dp"
        android:text="Student Details"
        android:textSize="30sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_x="20dp"
        android:layout_y="110dp"
        android:text="Enter Rollno:"
        android:textSize="20sp" />

    <EditText
        android:id="@+id/Rollno"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="175dp"
        android:layout_y="100dp"
        android:inputType="number" />

```

```
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
```

```
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />

<Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:text="View"
    android:textSize="30dp" />

<Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />

</AbsoluteLayout>
```

MainActivity.java

```
package com.example.db;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
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;

public class MainActivity extends Activity implements OnClickListener
{
    EditText Rollno,Name,Marks;
    Button Insert,Delete,Update,View,ViewAll;
    SQLiteDatabase db;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Rollno=(EditText)findViewById(R.id.Rollno);
        Name=(EditText)findViewById(R.id.Name);
        Marks=(EditText)findViewById(R.id.Marks);
        Insert=(Button)findViewById(R.id.Insert);
        Delete=(Button)findViewById(R.id.Delete);
        Update=(Button)findViewById(R.id.Update);
        View=(Button)findViewById(R.id.View);
        ViewAll=(Button)findViewById(R.id.ViewAll);

        Insert.setOnClickListener(this);
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);
    }
}
```

```

// Creating database and table
db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name VARCHAR,marks
VARCHAR);");
}

public void onClick(View view)
{
    // Inserting a record to the Student table
    if(view==Insert)
    {
        // Checking for empty fields
        if(Rollno.getText().toString().trim().length()==0|| 
           Name.getText().toString().trim().length()==0|| 
           Marks.getText().toString().trim().length()==0)
        {
            showMessage("Error", "Please enter all values");
            return;
        }
        db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+
                  "','"+"+Marks.getText()+"');");
        showMessage("Success", "Record added");
        clearText();
    }
    // Deleting a record from the Student table
    if(view==Delete)
    {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
            showMessage("Error", "Please enter Rollno");
            return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'",
                           null);
        if(c.moveToFirst())
        {
            db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
            showMessage("Success", "Record Deleted");
        }
        else
        {

```

```

        showMessage("Error", "Invalid Rollno");
    }
    clearText();
}
// Updating a record in the Student table
if(view==Update)
{
    // Checking for empty roll number
    if(Rollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
        return;
    }
    Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
    if(c.moveToFirst())
    {
        db.execSQL("UPDATE student SET name='"+Name.getText()+"',marks='"+ Marks.getText() +
                    "' WHERE rollno='"+Rollno.getText()+"'");
        showMessage("Success", "Record Modified");
    }
    else {
        showMessage("Error", "Invalid Rollno");
    }
    clearText();
}
// Display a record from the Student table
if(view==View)
{
    // Checking for empty roll number
    if(Rollno.getText().toString().trim().length()==0)
    {
        showMessage("Error", "Please enter Rollno");
        return;
    }
    Cursor c=db.rawQuery("SELECT * FROM student WHERE rollno='"+Rollno.getText()+"'", null);
    if(c.moveToFirst())
    {
        Name.setText(c.getString(1));
        Marks.setText(c.getString(2));
    }
}

```

```
        }
    else
    {
        showMessage("Error", "Invalid Rollno");
        clearText();
    }
}

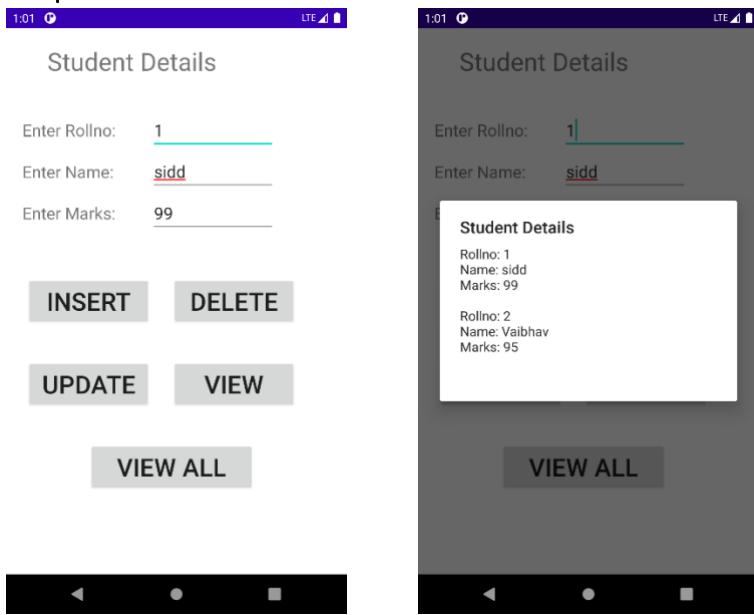
// Displaying all the records
if(view==ViewAll)
{
    Cursor c=db.rawQuery("SELECT * FROM student", null);
    if(c.getCount()==0)
    {
        showMessage("Error", "No records found");
        return;
    }
    StringBuffer buffer=new StringBuffer();
    while(c.moveToNext())
    {
        buffer.append("Rollno: "+c.getString(0)+"\n");
        buffer.append("Name: "+c.getString(1)+"\n");
        buffer.append("Marks: "+c.getString(2)+"\n\n");
    }
    showMessage("Student Details", buffer.toString());
}
}

public void showMessage(String title,String message)
{
    Builder builder=new Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}

public void clearText()
{
    Rollno.setText("");
    Name.setText("");
    Marks.setText("");
    Rollno.requestFocus();
}
```

}

Output:



Practical No.27

Create sample application with login module. (Check username and password) On successful login, Change Text View “Login Successful” and on login fail, alert user using Toast “Login fail”.

35 Siddharth Revankar TYCO

I. Practical Significance

This day Login and Registration form in Android are part of every application. So, when we are programming, we work with many registration forms. Forms can be very different from a simple login or registration to a complex ordering form for any application.

VII. Minimum Theoretical Background

A login application is the screen asking your credentials to login to some particular application. You might have seen it when logging into Facebook, twitter etc. First you have to define two Text View asking username and password of the user. Define a button with login text. In the java file, inside the method of onClick get the username and passwords text 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.

IX. Practical Related Questions

1. Explain the use of equals()function.

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

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

Ans.

Important functions which are related to GUI component “Button”:

- **onClick()**
- **setEnabled()**
- **setText()**
- **setVisibility()**
- **getText()**

3. State the uses of Toast message.

Ans. A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout.

X. Exercise

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

Ans.

Activity_main.xml

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

    <TextView
        android:id="@+id/textview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:background="#2A0E77"
        android:text="Login Page"
        android:textColor="#FFFFFF"
        android:textSize="35dp" />

    <EditText
        android:layout_width = "wrap_content"
        android:layout_height = "wrap_content"
        android:id = "@+id/editText"
        android:hint = "Enter Name"
        android:focusable = "true"
        android:layout_marginTop = "46dp"
        android:layout_below = "@+id/textview"
        android:layout_alignParentLeft = "true"
        android:layout_alignParentStart = "true"
        android:layout_alignParentRight = "true"
        android:layout_alignParentEnd = "true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
```

```
    android:ems="10"
    android:id="@+id/editText2"
    android:layout_below="@+id/editText"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText"
    android:layout_alignEnd="@+id/editText"
    android:hint="Password" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Attempts Left:"
    android:id="@+id/textView2"
    android:layout_below="@+id/editText2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="20dp" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/textView2"
    android:layout_alignBottom="@+id/textView2"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="-1dp"
    android:layout_marginEnd="70dp"
    android:layout_marginRight="70dp"
    android:layout_marginBottom="1dp"
    android:visibility="visible" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginTop="45dp"
    android:layout_marginEnd="-142dp"
    android:layout_marginRight="-142dp"
    android:layout_toStartOf="@+id/textview"
    android:layout_toLeftOf="@+id/textview"
```

```

        android:background="#2A0E77"
        android:text="login"
        android:textColor="#FFFFFF" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginStart="-146dp"
    android:layout_marginLeft="-146dp"
    android:layout_marginTop="45dp"
    android:layout_toEndOf="@+id/textview"
    android:layout_toRightOf="@+id/textview"
    android:background="#2A0E77"
    android:text="Cancel"
    android:textColor="#FFFFFF" />

</RelativeLayout>

```

MainActivity.java

```

package com.example.prac27;
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

public class MainActivity extends Activity {
    Button b1,b2;
    EditText ed1,ed2;

    TextView tx1;
    int counter = 3;

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

        b1 = (Button)findViewById(R.id.button);
        ed1 = (EditText)findViewById(R.id.editText);
        ed2 = (EditText)findViewById(R.id.editText2);
    }
}

```

```

b2 = (Button)findViewById(R.id.button2);
tx1 = (TextView)findViewById(R.id.textView3);
tx1.setVisibility(View.GONE);

b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ed1.getText().toString().equals("sidd") &&
           ed2.getText().toString().equals("abc123")) {
            Toast.makeText(getApplicationContext(),"Login
Successful",Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(getApplicationContext(), "Wrong
Credentials",Toast.LENGTH_SHORT).show();

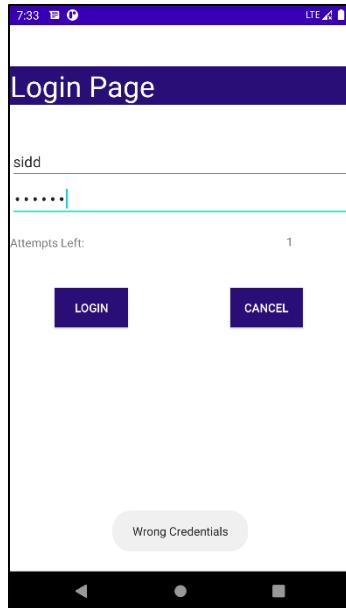
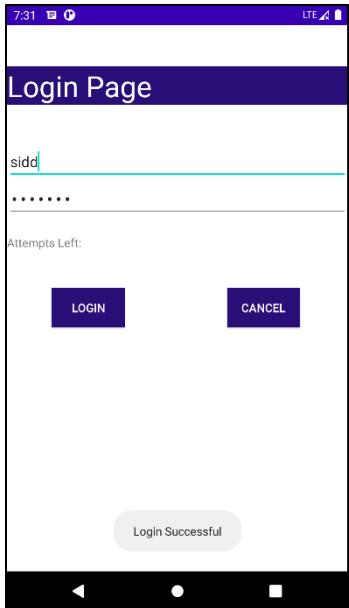
            tx1.setVisibility(View.VISIBLE);
            counter--;
            tx1.setText(Integer.toString(counter));

            if (counter == 0) {
                b1.setEnabled(false);
            }
        }
    });
});

b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}

```

Output:



Practical No.28

Create login application where you will have to validate username and password till the username and password is not validated, login button should remain disabled.

35 Siddharth Revankar TYCO

I. Practical Significance

This day Login and Registration form in Android are part of every application out there. So, when we are programming, we work with many registration forms. Forms can be very different from a simple login or registration to a complex ordering form for any application.

VII. Minimum Theoretical Background

With registration, how you can check data that the user has entered with simple validation. Validation can check many conditions. We can verify if an email address is a valid email and if a user entered all the required data, for instance, we check if Edit Text is empty for the first and last name. We can prepare a way to notify the user that the data is not valid. On login activity, we should check for password length. There are a few things login and registration form need:

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).

IX. Practical Related Questions

1. Explain validation of user input?

Ans.

Validation is a process to ensure that the value entered by the user is within the accepted boundaries of the application. 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. List and explain various GUI components used to design the login form with validation.

Ans.

GUI components used to design the login form:

- **TextView:** To display Login form name
- **EditText:** To get input from user
- **Button:** To submit login form

3. Differentiate between Text View and Edit Text View.

Ans.

TextView is used to display text and is not editable by the user. TextView can be updated programatically at any time.

EditText is used for user input. It is editable by the user. EditText control is an extended version of TextView control with additional features.

X. Exercise

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.

Ans.

Activity_main.xml

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

```
<TextView
    android:id="@+id/textview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
    android:background="#2A0E77"
    android:text="Login Page"
    android:textColor="#FFFFFF"
    android:textSize="35dp" />
```

```
<EditText
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:id = "@+id/editText"
    android:hint = "Enter Name"
    android:focusable = "true"
    android:layout_marginTop = "46dp"
    android:layout_below = "@+id/textview"
    android:layout_alignParentLeft = "true"
    android:layout_alignParentStart = "true"
    android:layout_alignParentRight = "true"
    android:layout_alignParentEnd = "true" />
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:ems="10"
    android:id="@+id/editText2"
    android:layout_below="@+id/editText"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignRight="@+id/editText"
    android:layout_alignEnd="@+id/editText"
    android:hint="Password" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Attempts Left:"
    android:id="@+id/textView2"
    android:layout_below="@+id/editText2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="20dp" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/textView2"
    android:layout_alignBottom="@+id/textView2"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_marginTop="-1dp"
    android:layout_marginEnd="70dp"
    android:layout_marginRight="70dp"
    android:layout_marginBottom="1dp"
    android:visibility="visible" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginTop="45dp"
```

```

        android:layout_marginEnd="-142dp"
        android:layout_marginRight="-142dp"
        android:layout_toStartOf="@+id/textview"
        android:layout_toLeftOf="@+id/textview"
        android:background="#2A0E77"
        android:text="login"
        android:textColor="#FFFFFF" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView2"
    android:layout_marginStart="-146dp"
    android:layout_marginLeft="-146dp"
    android:layout_marginTop="45dp"
    android:layout_toEndOf="@+id/textview"
    android:layout_toRightOf="@+id/textview"
    android:background="#2A0E77"
    android:text="Cancel"
    android:textColor="#FFFFFF" />

</RelativeLayout>

```

MainActivity.java

```

package com.example.prac28;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.*;

public class MainActivity extends Activity {
    Button b1,b2;
    EditText ed1,ed2;
    TextView tx1;
    int counter = 3;

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

        b1 = (Button)findViewById(R.id.button);

```

```

ed1 = (EditText)findViewById(R.id.editText);
ed2 = (EditText)findViewById(R.id.editText2);

b2 = (Button)findViewById(R.id.button2);
tx1 = (TextView)findViewById(R.id.textView3);
tx1.setVisibility(View.GONE);

b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (ed1.getText().toString().isEmpty()) {
            Toast.makeText(getApplicationContext(),"Enter user
name!!",Toast.LENGTH_SHORT).show();
        }
        if (ed2.getText().toString().isEmpty()) {
            Toast.makeText(getApplicationContext(),"Enter
password!!",Toast.LENGTH_SHORT).show();
        }
        if (ed1.getText().toString().length() < 5) {
            Toast.makeText(getApplicationContext(),"User name should be more than 5
letters",Toast.LENGTH_SHORT).show();
        }
        if (ed2.getText().toString().length() < 6) {
            Toast.makeText(getApplicationContext(),"Password should be more than 6
letters",Toast.LENGTH_SHORT).show();
        }
        if(ed1.getText().toString().equals("siddharth") &&
           ed2.getText().toString().equals("sidd123")) {
            Toast.makeText(getApplicationContext(),"Login
Successful",Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(getApplicationContext(), "Login
unsuccessful",Toast.LENGTH_SHORT).show();
        }

        tx1.setVisibility(View.VISIBLE);
        counter--;
        tx1.setText(Integer.toString(counter));

        if (counter == 0) {
            b1.setEnabled(false);
        }
    }
}

```

```

});
```

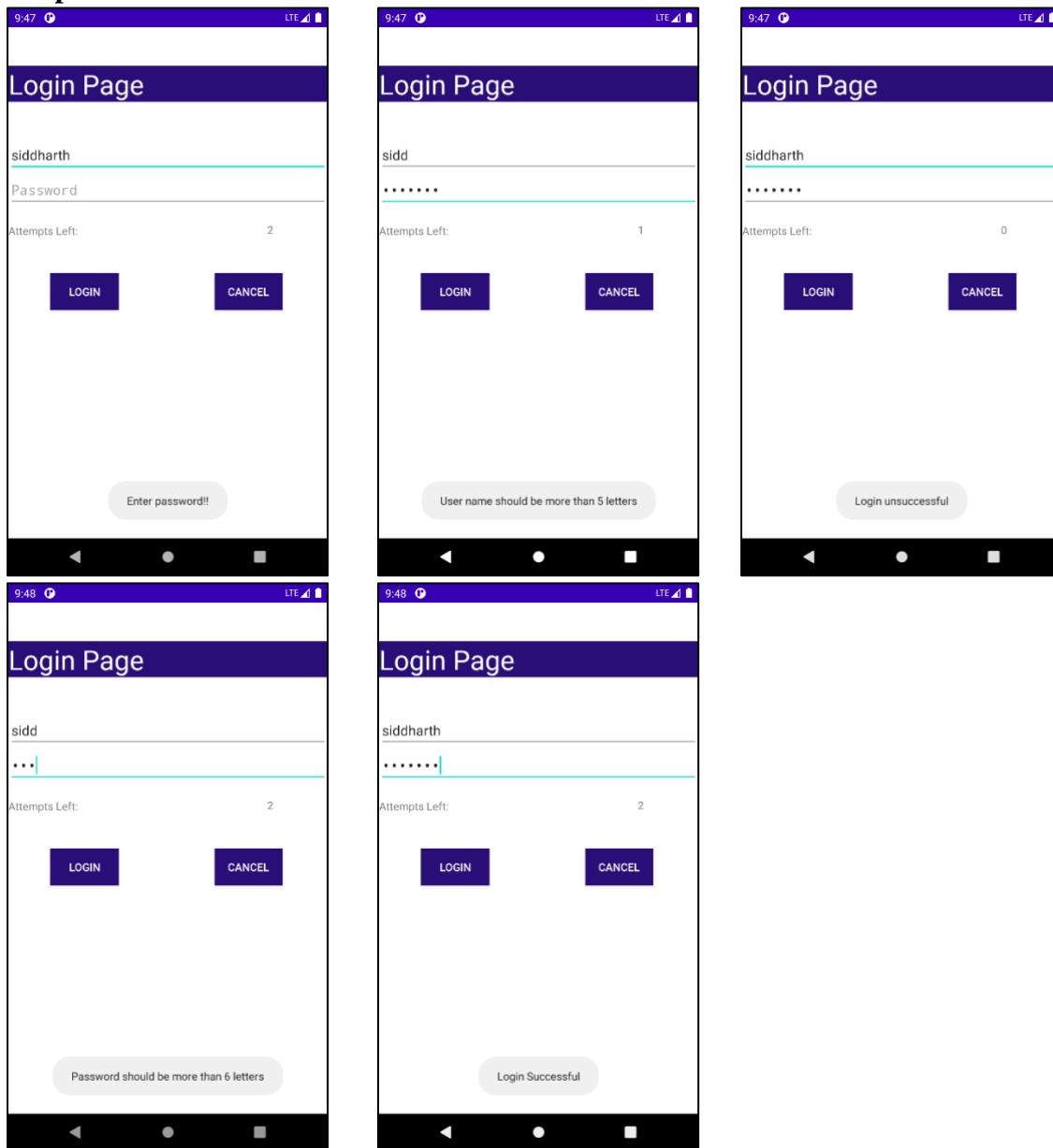
```

b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
```

```

}
}
```

Output:



Practical No.29

Develop a program to: a) Send SMS b) Receive SMS

35 Siddharth RevankarTYCO

I. Practical Significance

Android devices can send and receive messages to or from any other phone that supports Short Message Service (SMS). Android offers the Messenger application that can send and receive SMS messages.

VII. Minimum Theoretical Background

A host of third-party apps for sending and receiving SMS messages are also available in Google Play. The SMS protocol was primarily designed for user-to-user communication and is not well-suited for apps that want to transfer data. You should not use SMS to send data messages from a web server to your app on a user device. SMS is neither encrypted nor strongly authenticated on either the network or the device. Access to the SMS features of an Android device is protected by user permissions. Just as your app needs the user's permission to use phone features, so also does an app need the user's permission to directly use SMS features. You have two choices for *sending* SMS messages:

- Use an implicit Intent to launch a messaging app such as Messenger, with the ACTION_SENDTO action.
- Send the SMS message using the sendTextMessage() method or other methods of the SmsManager class.

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 destinationAddress, 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.

IX. Practical Related Questions

1. Explain the use of SmsManagerClass.

Ans. Manages SMS operations such as sending data, text, and pdu SMS messages. Get this object by calling the static method SmsManager.getDefault().

2. List changes that are need to be done in AndroidManifest.XML file to send and receive messages.

Ans. We need to give 2 permissions in the file to send and receive messages

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

X. Exercise

1. Write a program to send and receive SMS, make use of the following GUI.



Ans.

Activity_main.xml

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

```
    android:layout_height="match_parent"
    tools:context="MainActivity">

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="117dp"
    android:layout_marginLeft="117dp"
    android:layout_marginTop="108dp"
    android:text="Sending SMS"
    android:textSize="30dp" />

<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:hint="Enter Phone Number"
    android:phoneNumber="true" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:layout_below="@+id/editText"
    android:layout_alignLeft="@+id/editText"
    android:layout_alignStart="@+id/editText"
    android:layout_marginTop="20dp"
    android:layout_alignRight="@+id/textView1"
    android:layout_alignEnd="@+id/textView1"
    android:hint="Enter SMS" />

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

```
    android:layout_below="@+id/editText2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
    android:text="Send Sms" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.prac29;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.app.Activity;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends Activity {
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 0 ;
    Button sendBtn;
    EditText txtphoneNo;
    EditText txtMessage;
    String phoneNo;
    String message;

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

        sendBtn = (Button) findViewById(R.id.btnSendSMS);
        txtphoneNo = (EditText) findViewById(R.id.editText);
        txtMessage = (EditText) findViewById(R.id.editText2);

        sendBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sendSMSMessage();
            }
        });
    }

    private void sendSMSMessage() {
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.SEND_SMS) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this,
                    new String[]{Manifest.permission.SEND_SMS}, MY_PERMISSIONS_REQUEST_SEND_SMS);
        } else {
            String strPhoneNo = txtphoneNo.getText().toString();
            String strMessage = txtMessage.getText().toString();

            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(strPhoneNo, null, strMessage, null, null);
        }
    }
}
```

```

        }
    });
}

protected void sendSMSMessage() {
    phoneNo = txtphoneNo.getText().toString();
    message = txtMessage.getText().toString();

    if (ContextCompat.checkSelfPermission(this,
            Manifest.permission.SEND_SMS)
        != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                Manifest.permission.SEND_SMS)) {
        } else {
            ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.SEND_SMS},
                MY_PERMISSIONS_REQUEST_SEND_SMS);
        }
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, String permissions[], int[]
grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_SEND_SMS: {
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                SmsManager smsManager = SmsManager.getDefault();
                smsManager.sendTextMessage(phoneNo, null, message, null, null);
                Toast.makeText(getApplicationContext(), "SMS sent.",
                    Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText(getApplicationContext(), "SMS failed, please try again.",
                    Toast.LENGTH_LONG).show();
                return;
            }
        }
    }
}

```

IncomingSms.java

```
package com.example.prac29;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.telephony.SmsMessage;
import android.util.Log;
import android.widget.Toast;

public class IncomingSms extends BroadcastReceiver {

    // Get the object of SmsManager
    final SmsManager sms = SmsManager.getDefault();

    public void onReceive(Context context, Intent intent) {

        // Retrieves a map of extended data from the intent.
        final Bundle bundle = intent.getExtras();

        try {

            if (bundle != null) {

                final Object[] pdusObj = (Object[]) bundle.get("pdus");

                for (int i = 0; i < pdusObj.length; i++) {

                    SmsMessage currentMessage = SmsMessage.createFromPdu((byte[]) pdusObj[i]);
                    String phoneNumber = currentMessage.getDisplayOriginatingAddress();

                    String senderNum = phoneNumber;
                    String message = currentMessage.getDisplayMessageBody();

                    Log.i("SmsReceiver", "senderNum: " + senderNum + "; message: " + message);

                    // Show Alert
                    int duration = Toast.LENGTH_LONG;
                    Toast toast = Toast.makeText(context,
                            "senderNum: " + senderNum + ", message: " + message, duration);
                    toast.show();
                }
            }
        }
    }
}
```

```

        } // end for loop
    } // bundle is null

} catch (Exception e) {
    Log.e("SmsReceiver", "Exception smsReceiver" +e);

}
}

}
}

```

AndroidManifest.xml:

```

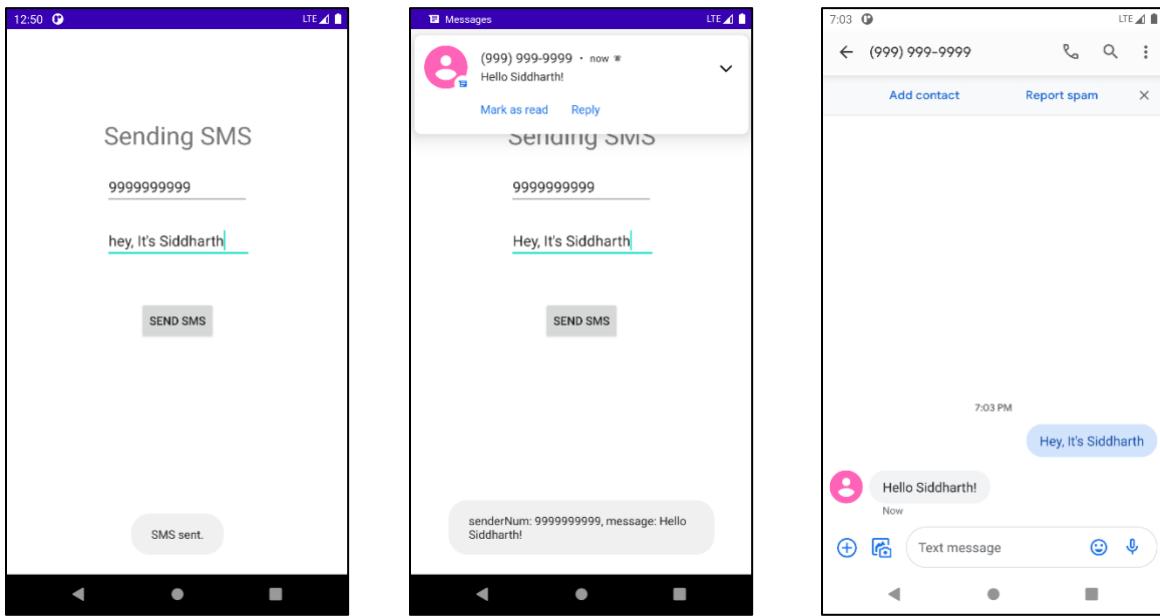
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.prac29">
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <uses-permission android:name="android.permission.READ_SMS" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Prac29">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".IncomingSms">
            <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
    </application>

</manifest>

```

Output:



Practical No.30
Develop a program to send and receive e-mail
35 Siddharth Revankar TYCO

I. Practical Significance

To send email from your Android application, you have to write an Activity that needs to launch an email client and sends an email using your Android device. This practical focuses on integrating the existing email clients in the new applications

VII. Minimum Theoretical Background

Intent Object - Action to send Email:

You will use ACTION_SEND action to launch an email client installed on your Android device. Following is simple syntax to create an intent with ACTION_SEND action.

```
Intent emailIntent = new Intent(Intent.ACTION_SEND);
```

Intent Object – Data Type to send Email

To send an email you need to specify mailto: as URI using setData() method and data type will be to text/plain using setType() method as follows:

```
emailIntent.setData(Uri.parse("mailto:"));  
emailIntent.setType("text/plain");
```

IX. Practical Related Questions

1. Why it becomes necessary to have inbuilt email module in mobile applications.

Ans. To send an email from our application, we don't have to implement an email client from the beginning, but we can use an existing one like the default Email app provided from Android.

2. List the extra fields that can be used in an application to send emails.

Ans.

a) EXTRA_BCC

A String[] holding e-mail addresses that should be blind carbon copied.

b) EXTRA_CC

A String[] holding e-mail addresses that should be carbon copied.

c) EXTRA_EMAIL

A String[] holding e-mail addresses that should be delivered to.

d) EXTRA_HTML_TEXT

A constant String that is associated with the Intent, used with ACTION_SEND to supply an alternative to EXTRA_TEXT as HTML formatted text.

e) EXTRA_SUBJECT

A constant string holding the desired subject line of a message.

f) EXTRA_TEXT

A constant CharSequence that is associated with the Intent, used with ACTION_SEND to supply the literal data to be sent.

g) EXTRA_TITLE

A CharSequence dialog title to provide to the user when used with a ACTION_CHOOSER.

X. Exercise

1. Write a program to send email.

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center"
        android:padding="20dp"
        tools:context=".MainActivity">
```

```
<TextView
```

```
    android:id="@+id/textView"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="112dp"
    android:text="To"
    android:textSize="20sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
```

```
    android:id="@+id/et_To"
    android:layout_width="260dp"
    android:layout_height="40dp"
    android:layout_marginTop="108dp"
    android:layout_marginEnd="20dp"
    android:layout_marginRight="20dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.867"
    app:layout_constraintStart_toEndOf="@+id/textView"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:id="@+id/et_subject"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp"
    android:hint="Subject"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.7"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/et_To" />
```

```
<EditText
    android:id="@+id/et_Message"
    android:layout_width="300dp"
    android:layout_height="233dp"
    android:gravity="top"
    android:hint="Message"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.531"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/et_subject"
    app:layout_constraintVertical_bias="0.125" />
```

```
<Button
    android:id="@+id/bt_Send"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="520dp"
    android:text="Send"
    android:textSize="20sp"
    app:backgroundTint="#3F51B5"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="157dp"
```

```
    android:layout_height="37dp"
    android:layout_marginTop="28dp"
    android:background="#E8E5E5"
    android:gravity="center"
    android:text="Practical 30"
    android:textColor="#673AB7"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

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

MainActivity.java

```
package com.example.practical30;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    private EditText eTo;
    private EditText etSubject;
    private EditText etMesssage;
    Button btSend;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        eTo = findViewById(R.id.et_To);
        etMesssage = findViewById(R.id.et_Message);
        etSubject = findViewById(R.id.et_subject);
        btSend = findViewById(R.id.bt_Send);
        btSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        sendMail();
    }
});
}
}

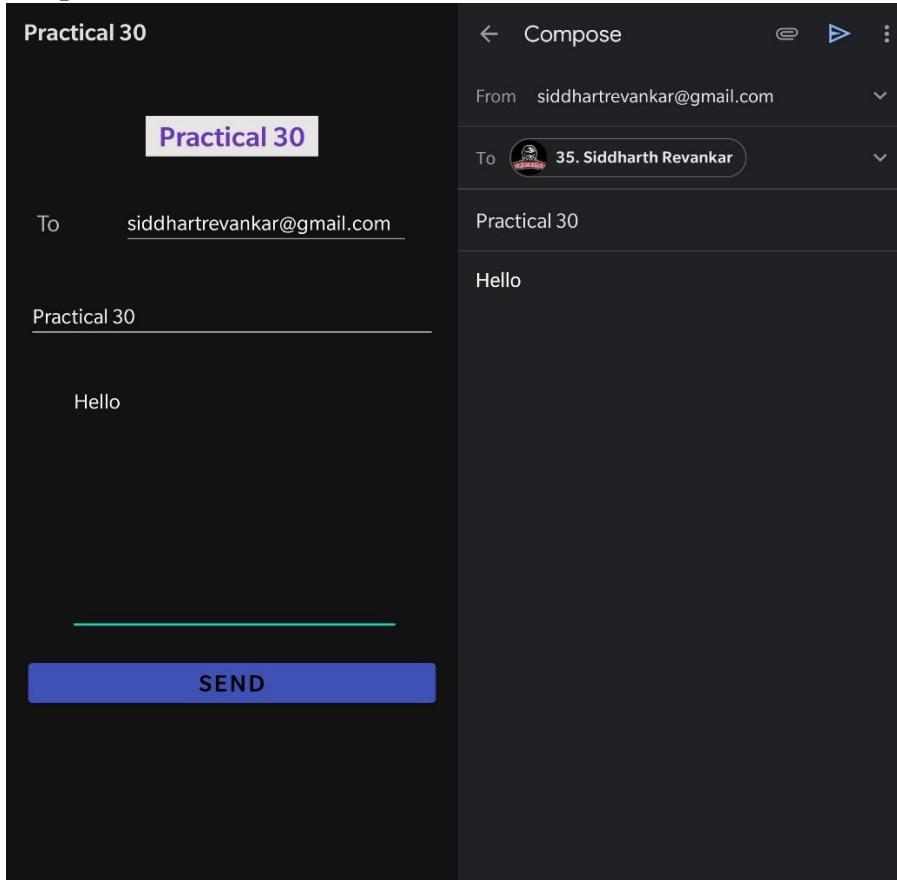
private void sendMail() {
    String recipientList = eTo.getText().toString();
    String[] recipients = recipientList.split(",");
    String subject = etSubject.getText().toString();
    String message = etMessage.getText().toString();

    Intent intent = new Intent(Intent.ACTION_SEND);
    intent.putExtra(Intent.EXTRA_SUBJECT,subject);
    intent.putExtra(Intent.EXTRA_EMAIL,recipients);
    intent.putExtra(Intent.EXTRA_TEXT,message);

    intent.setType("message/rfc822");
    startActivity(Intent.createChooser(intent,"Choose email Client"));
}
}

```

Output:



Practical No.31
Deploy map-based application. Part I
35 Siddharth Revankar TYCO

I. Practical Significance

Android allows us to integrate google maps in our application. You can show any location on the map, or can show different routes on the map. You can also customize the map according to your choices.

VII. Minimum Theoretical Background

Add the map fragment into xml layout file. Its syntax is given below –

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

Add some permissions along with the Google Map

API key in the AndroidManifest.XML file. Its syntax is given below:

```
<!--Permissions-->  
<uses-permission  
    android:name="android.permission.ACCESS_NETWORK_STATE"  
/>  
  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission  
    android:name="com.google.android.providers.gsf.permission.  
READ_GSERVICES" />  
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STO  
RAGE" />
```

IX. Practical Related Questions

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

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

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

Syntax to Change:

1. googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
2. googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
3. googleMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
4. googleMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);

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

Ans. Methods of zoom enable and disable feature:

setIsZoomInEnabled(boolean isEnabled)

X. Exercise

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

Ans.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:layout_width="match_parent"
        android:layout_height="65dp"
        android:id="@+id/button4"
        android:layout_alignParentBottom="true"
        android:backgroundTint="#00D100"
        android:onClick="btnCurrentLocation"
        android:text="Current Location"
        android:textColor="#FFFFFF"
        android:layout_margin="30dp"
        android:textSize="22sp"
    />
</RelativeLayout>
```

MainActivity.java

```
package com.example.practical31;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
public class MainActivity extends AppCompatActivity {
    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
public void btnCurrentLocation(View view)
{
    startActivity(new Intent(this, MapsActivity.class));
}
}
```

MapsActivity.java:

```
package com.example.practical31;

import android.Manifest;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.provider.Settings;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.karumi.dexter.Dexter;
```

```

import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.single.PermissionListener;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
        GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener ,
        com.google.android.gms.location.LocationListener {

    private GoogleMap mMap;
    private GoogleApiClient mGoogleApiClient;
    private Location mLocation;
    private LocationManager mLocationManager;
    private LocationRequest mLocationRequest;
    private com.google.android.gms.location.LocationListener listener;
    private long UPDATE_INTERVAL = 2000;
    private long FASTEST_INTERVAL = 5000;
    private LocationManager locationManager;
    private LatLng latLan;
    private boolean isPermission;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        if(requestSinglePermissions()) {// Obtain the SupportMapFragment and get notified when
the map is ready to be used.
            SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
            mapFragment.getMapAsync(this);

            mGoogleApiClient = new GoogleApiClient.Builder(this)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .addApi(LocationServices.API)
                .build();
            mLocationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
            checkLocation();
        }
    }
    private boolean checkLocation() {
        if(!isLocationEnabled()){

```

```

        showAlert();
    }
    return isLocationEnabled();
}
private boolean isLocationEnabled() {
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);
}
private void showAlert(){
    final AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("Enable Location")
        .setMessage("Your Location Settings is set to off. \n Please enable Location to "+"to
use this app")
        .setPositiveButton("Location Settings", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                Intent myIntent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                startActivity(myIntent);
            }
        })
        .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
            }
        });
    dialog.show();
}
private boolean requestSinglePermissions() {
    Dexter.withActivity(this)
        .withPermission(Manifest.permission.ACCESS_FINE_LOCATION)
        .withListener(new PermissionListener() {
            @Override
            public void onPermissionGranted(PermissionGrantedResponse response) {
                isPermission = true;
            }
            @Override
            public void onPermissionDenied(PermissionDeniedResponse response) {
                isPermission = false;
            }
            @Override
            public void onPermissionRationaleShouldBeShown(PermissionRequest permission,

```

```

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

    if(latLan!=null)
    {
        mMap.addMarker(new MarkerOptions().position(latLan).title("Marker in Current
Location"));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLan, 14F));
    }
}
@Override
public void onConnected(@Nullable Bundle bundle) {
    if(ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)!=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED){
        return;
    }
    startLocationUpdates();
    mLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
    if(mLocation == null)
    {
        startLocationUpdates();
    }
    else
    {
        Toast.makeText(this, "Location not detected", Toast.LENGTH_SHORT).show();
    }
}

```

```

        }
    }
    private void startLocationUpdates() {

        mLocationRequest = LocationRequest.create()
            .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
            .setInterval(UPDATE_INTERVAL)
            .setFastestInterval(FASTEST_INTERVAL);
        if(ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED){
            return;
        }
        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
    }

    @Override
    public void onConnectionSuspended(int i) {
    }

    @Override
    public void onLocationChanged(Location location) {
        String msg = "Updated Location: "+ Double.toString(location.getLatitude())+",
"+Double.toString(location.getLongitude());
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
        latLan = new LatLng(location.getLatitude(), location.getLongitude());
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    }

    @Override
    protected void onStart() {
        super.onStart();
        if(mGoogleApiClient != null)
        {
            mGoogleApiClient.connect();
        }
    }

    @Override

```

```
protected void onStop() {
    super.onStop();
    if(mGoogleApiClient.isConnected())
    {
        mGoogleApiClient.disconnect();
    }
}
build.gradle(:app):
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.practical31"
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-
rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures {
        viewBinding true
    }
}
```

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.android.gms:play-services-maps:17.0.1'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.karumi:dexter:6.2.2'
    implementation 'com.google.android.gms:play-services-maps:17.0.1'
    implementation 'com.google.android.gms:play-services-location:18.0.0'
}
```

google_maps_api.xml:

```
<resources>
```

```
<!--
```

TODO: Before you run your application, you need a Google Maps API key.

To get one, follow this link, follow the directions and press "Create" at the end:

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=87:6A:99:09:31:A0:CC:64:84:CD:62:00:17:C6:13:16:5A:A6:8C:3B%3Bcom.example.practical31

You can also add your credentials to an existing key, using these values:

Package name:

com.example.practical31

SHA-1 certificate fingerprint:

87:6A:99:09:31:A0:CC:64:84:CD:62:00:17:C6:13:16:5A:A6:8C:3B

Alternatively, follow the directions here:

<https://developers.google.com/maps/documentation/android/start#get-key>

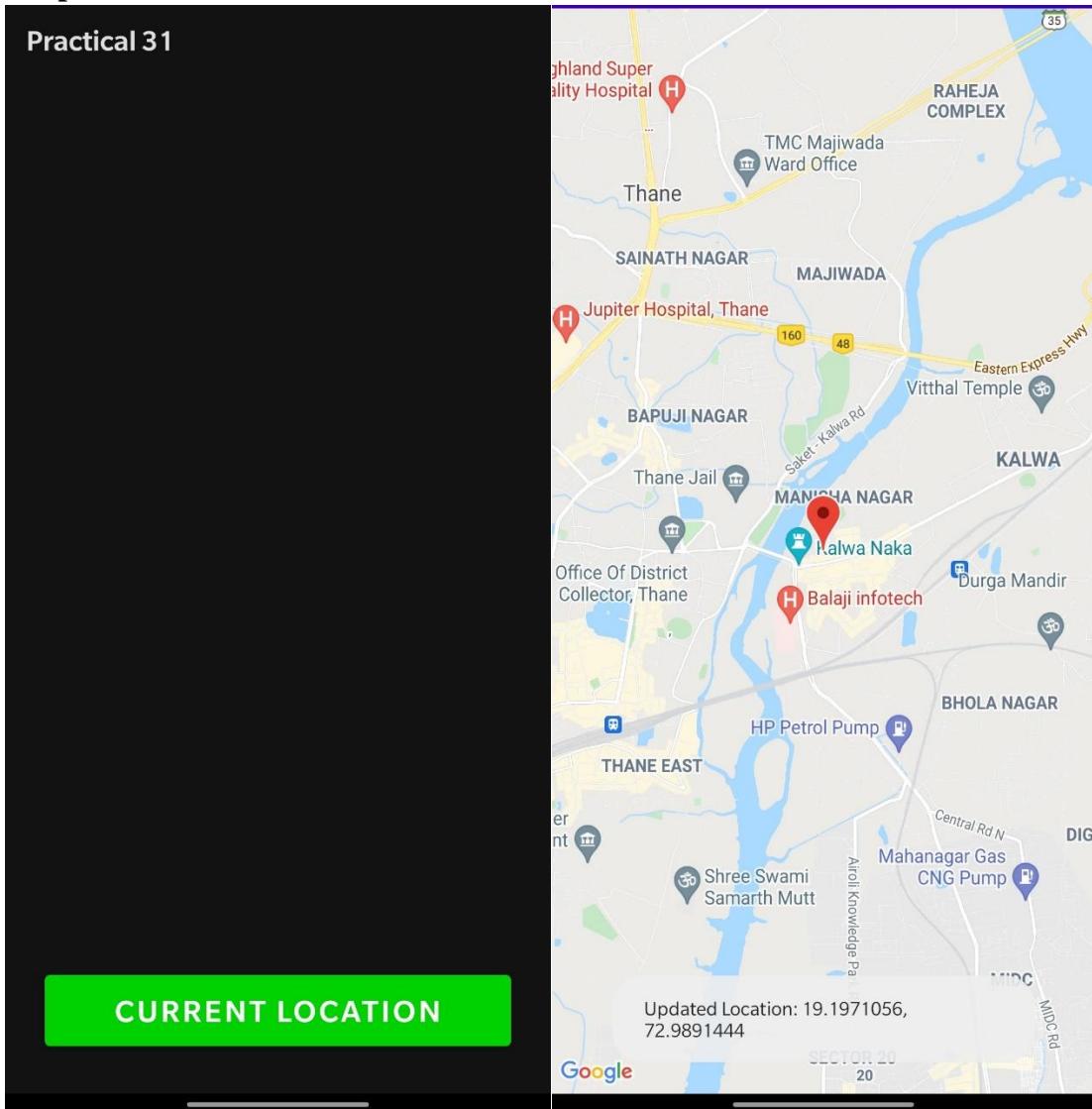
Once you have your key (it starts with "AIza"), replace the "google_maps_key" string in this file.

-->

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyCBHsZXnpHjSIxkQxplPbr-9XIIMZpWyJI</string>
</resources>
```

Output:

Practical 31



Practical No.32
Deploy map-based application. Part II
35 Siddharth Revankar TYCO

I. Practical Significance

Android allows us to integrate google maps in our application. You can show any location on the map, or can show different routes on the map. You can also customize the map according to your choices.

VII. Minimum Theoretical Background

Methods available in the Google Map class are given below.

1. addCircle(CircleOptions options) : This method add a circle to the map
2. addPolygon(PolygonOptions options) : This method add a polygon to the map
3. addTileOverlay(TileOverlayOptions options) : This method add tile overlay to the map
4. animateCamera(CameraUpdate update) : This method Moves the map according to the update with an animation
5. clear() : This method removes everything from the map
6. getMyLocation() : This method returns the currently displayed user location
7. moveCamera(CameraUpdate update) : This method repositions the camera according to the instructions defined in the update
8. setTrafficEnabled(boolean enabled) : This method Toggles the traffic layer on or off
9. snapshot(GoogleMap.SnapshotReadyCallback callback) : This method Takes a snapshot of the map
10. stopAnimation() : This method stops the camera animation if there is one in progress

IX. Practical Related Questions

1. Explain the ways to add Markers on the Google Map

Ans. For adding a custom marker to Google Maps navigate to the app > res > drawable > Right- Click on it > New > Vector Assets and select the icon which we have to show on your Map. You can change the color according to our requirements. After creating this icon now we will move towards adding this marker to our Map.

2. Write the syntax for method which is used to add compass in Google Map.

Ans. You can set compass by calling below method:

```
mMap. getUiSettings(). setCompassEnabled(true);
```

X. Exercise

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

Ans.

Activity_maps.xml:

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

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

DirectionsJSONParser.java:

```
package com.example.practical32;
```

```
import java.util.ArrayList;
```

```
import java.util.HashMap;
```

```
import java.util.List;
```

```
import org.json.JSONArray;
```

```
import org.json.JSONException;
```

```
import org.json.JSONObject;
```

```
import com.google.android.gms.maps.model.LatLng;
```

```
public class DirectionsJSONParser {
```

```
    public List<List<HashMap<String, String>>> parse(JSONObject jObject){
```

```
        List<List<HashMap<String, String>>> routes = new ArrayList<>();
```

```
        JSONArray jRoutes = null;
```

```
JSONArray jLegs = null;
JSONArray jSteps = null;

try {

    jRoutes = jObject.getJSONArray("routes");
    for(int i=0;i<jRoutes.length();i++){
        jLegs = ( (JSONObject)jRoutes.get(i)).getJSONArray("legs");
        List<HashMap<String, String>> path = new ArrayList<HashMap<String, String>>();
        for(int j=0;j<jLegs.length();j++){
            jSteps = ( (JSONObject)jLegs.get(j)).getJSONArray("steps");
            for(int k=0;k<jSteps.length();k++){
                String polyline = "";
                polyline =
                (String)((JSONObject)((JSONObject)jSteps.get(k)).get("polyline")).get("points");
                List<LatLng> list = decodePoly(polyline);
                for(int l=0;l<list.size();l++){
                    HashMap<String, String> hm = new HashMap<String, String>();
                    hm.put("lat", Double.toString(((LatLng)list.get(l)).latitude) );
                    hm.put("lng", Double.toString(((LatLng)list.get(l)).longitude) );
                }
            }
        }
    }
}
```

```

        path.add(hm);

    }

    routes.add(path);

}

}

} catch (JSONException e) {

    e.printStackTrace();

}catch (Exception ignored){

}

return routes;

}

/**
 * Method to decode polyline points
 *
 * Courtesy : http://jeffreysambells.com/2010/05/27/decoding-polylines-from-google-maps-direction-api-with-java
 */
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 & 0x1f) << shift;

    shift += 5;

} while (b >= 0x20);

int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));

lat += dlat;

shift = 0;

result = 0;

do {

    b = encoded.charAt(index++) - 63;

    result |= (b & 0x1f) << shift;

    shift += 5;

} while (b >= 0x20);

int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));

lng += dlng;

LatLng p = new LatLng(((double) lat / 1E5),

                      (((double) lng / 1E5)));

final boolean add = poly.add(p);

}
```

```
    return poly;  
}  
}
```

MapsActivity.java:

```
package com.example.practical32;  
  
import android.graphics.Color;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.Toast;  
  
import androidx.fragment.app.FragmentActivity;  
  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.BitmapDescriptorFactory;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.google.android.gms.maps.model.Polyline;  
import com.google.android.gms.maps.model.PolylineOptions;  
  
import org.json.JSONObject;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Objects;  
  
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback{  
    private GoogleMap mMap;
```

```
private LatLng mOrigin;
private LatLng mDestination;
private Polyline mPolyline;
ArrayList<LatLng> mMarkerPoints;

@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_maps);

    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
    .findFragmentById(R.id.map);

    mapFragment.getMapAsync(this);

    mMarkerPoints = new ArrayList<>();
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
        @Override
        public void onMapClick(LatLng point) {
            // Already two locations
            if(mMarkerPoints.size()>1){
                mMarkerPoints.clear();
                mMap.clear();
            }
            // Adding new item to the ArrayList
            mMarkerPoints.add(point);

            // Creating MarkerOptions
            MarkerOptions options = new MarkerOptions();

            // Setting the position of the marker
            options.position(point);
        }
    });
}
```

```

        if(mMarkerPoints.size()==1){

options.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN))
;
        }else if(mMarkerPoints.size()==2){

options.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
        }

        // Add new marker to the Google Map Android API V2
        mMap.addMarker(options);

        // Checks, whether start and end locations are captured
        if(mMarkerPoints.size() >= 2){
            mOrigin = mMarkerPoints.get(0);
            mDestination = mMarkerPoints.get(1);
            drawRoute();
        }

    }
);
}
}

private void drawRoute(){

    // Getting URL to the Google Directions API
    String url = getDirectionsUrl(mOrigin, mDestination);

    DownloadTask downloadTask = new DownloadTask();

    // Start downloading json data from Google Directions API
    downloadTask.execute(url);
}

private String getDirectionsUrl(LatLng origin,LatLng dest){

    // Origin of route
    String str_origin = "origin="+origin.latitude+","+origin.longitude;

    // Destination of route
    String str_dest = "destination="+dest.latitude+","+dest.longitude;
}

```

```

// Key
String key = "key=" + getString(R.string.google_maps_key);

// Building the parameters to the web service
String parameters = str_origin+"&" +str_dest+"&" +key;

// Output format
String output = "json";

// Building the url to the web service
String url = "https://maps.googleapis.com/maps/api/directions/" +output+"?" +parameters;

return url;
}

/** A method to download json data from url */
private String downloadUrl(String strUrl) throws IOException {
    String data = "";
    InputStream iStream = null;
    HttpURLConnection urlConnection = null;
    try{
        URL url = new URL(strUrl);

        // Creating an http connection to communicate with url
        urlConnection = (HttpURLConnection) url.openConnection();

        // Connecting to url
        urlConnection.connect();

        // Reading data from url
        iStream = urlConnection.getInputStream();

        BufferedReader br = new BufferedReader(new InputStreamReader(iStream));

        StringBuffer sb = new StringBuffer();

        String line = "";
        while( ( line = br.readLine()) != null){
            sb.append(line);
        }

        data = sb.toString();
    }
}

```

```

        br.close();

    }catch(Exception e){
        Log.d("Exception on download", e.toString());
    }finally{
        iStream.close();
        urlConnection.disconnect();
    }
    return data;
}

/** A class to download data from Google Directions URL */
private class DownloadTask extends AsyncTask<String, Void, String> {

    // Downloading data in non-ui thread
    @Override
    protected String doInBackground(String... url) {

        // For storing data from web service
        String data = "";

        try{
            // Fetching the data from web service
            data = downloadUrl(url[0]);
            Log.d("DownloadTask","DownloadTask : " + data);
        }catch(Exception e){
            Log.d("Background Task",e.toString());
        }
        return data;
    }

    // Executes in UI thread, after the execution of
    // doInBackground()
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);

        ParserTask parserTask = new ParserTask();

        // Invokes the thread for parsing the JSON data
        parserTask.execute(result);
    }
}

```

```

}

/** A class to parse the Google Directions in JSON format */
private class ParserTask extends AsyncTask<String, Integer,
List<List<HashMap<String, String>>> >{

    // Parsing the data in non-ui thread
    @Override
    protected List<List<HashMap<String, String>>> doInBackground(String... jsonData) {

        JSONObject jObject;
        List<List<HashMap<String, String>>> routes = null;

        try{
            jObject = new JSONObject(jsonData[0]);
            DirectionsJSONParser parser = new DirectionsJSONParser();

            // Starts parsing data
            routes = parser.parse(jObject);
        }catch(Exception e){
            e.printStackTrace();
        }
        return routes;
    }

    // Executes in UI thread, after the parsing process
    @Override
    protected void onPostExecute(List<List<HashMap<String, String>>> result) {
        ArrayList<LatLng> points = null;
        PolylineOptions lineOptions = null;

        // Traversing through all the routes
        for(int i=0;i<result.size();i++){
            points = new ArrayList<LatLng>();
            lineOptions = new PolylineOptions();

            // Fetching i-th route
            List<HashMap<String, String>> path = result.get(i);

            // Fetching all the points in i-th route
            for(int j=0;j<path.size();j++){
                HashMap<String, String> point = path.get(j);
            }
        }
    }
}

```

```

        double lat = Double.parseDouble(Objects.requireNonNull(point.get("lat")));
        double lng = Double.parseDouble(Objects.requireNonNull(point.get("lng")));
        LatLng position = new LatLng(lat, lng);

        points.add(position);
    }

    // Adding all the points in the route to LineOptions
    lineOptions.addAll(points);
    lineOptions.width(8);
    lineOptions.color(Color.RED);
}

// Drawing polyline in the Google Map for the i-th route
if(lineOptions != null) {
    if(mPolyline != null){
        mPolyline.remove();
    }
    mPolyline = mMap.addPolyline(lineOptions);
}

}else
    Toast.makeText(getApplicationContext(),"No route is found",
Toast.LENGTH_LONG).show();
}
}
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.practical32">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"

```

```

        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Practical32">

    <!--
        The API key for Google Maps-based APIs is defined as a string resource.
        (See the file "res/values/google_maps_api.xml").
        Note that the API key is linked to the encryption key used to sign the APK.
        You need a different API key for each encryption key, including the release key that is
    used to
        sign the APK for publishing.
        You can define the keys for the debug and release targets in src/debug/ and src/release/.

    -->
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyCBHsZXnpHjSIxkQxplPbr-9XIlMZpWyJI" />

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

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

</manifest>
Google_maps_api:
<resources>
    <!--
        TODO: Before you run your application, you need a Google Maps API key.
    -->
```

To get one, follow this link, follow the directions and press "Create" at the end:

https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&r=87:6A:99:09:31:A0:CC:64:84:CD:62:00:17:C6:13:16:5A:A6:8C:3B%3Bcom.example.practical32

You can also add your credentials to an existing key, using these values:

Package name:
com.example.practical32

SHA-1 certificate fingerprint:
87:6A:99:09:31:A0:CC:64:84:CD:62:00:17:C6:13:16:5A:A6:8C:3B

Alternatively, follow the directions here:
<https://developers.google.com/maps/documentation/android/start#get-key>

Once you have your key (it starts with "AIza"), replace the "google_maps_key" string in this file.

```
-->
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyCBHsZXnpHjSIxkQxplPbr-9XIIMZpWyJI</string>
</resources>
```

Build.gradle(:app):

```
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.practical32"
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-
rules.pro'
        }
    }
    compileOptions {
```

```
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
buildFeatures {
    viewBinding true
}
}

dependencies {

implementation 'androidx.appcompat:appcompat:1.3.0'
implementation 'com.google.android.material:material:1.3.0'
implementation 'com.google.android.gms:play-services-maps:17.0.1'
implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.2'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
androidTestImplementation 'androidx.test:runner:1.3.0'
}
```

Output:

