EXPERIMENT 1

1. List different Android O. S. versions.

Cupcake	1.5	Lollipop	5.0 - 5.1.1
Donut	1.6	Marshmallow	6.0 - 6.0.1
Eclair	2.0 - 2.1	Nougat	7.0
Froyo	2.2 - 2.2.3	Nougat	7.1.0 - 7.1.2
Gingerbread	2.3 - 2.3.7	Oreo	8.0
Honeycomb	3.0 - 3.2.6	Oreo	8.1
Ice Cream Sandwich	4.0 - 4.0.4	Pie	9.0
Jelly Bean	4.1 - 4.3.1	Android 10	
KitKat	4.4 - 4.4.4	Android 11	

2. State characteristics of android operating sytsem
    1) Near Field Communication (NFC)
    2) Alternate Keyboards.
    3) Infrared Transmission.
    4) No-Touch Control.
    5) Automation.
    6) Wireless App Downloads.
    7) Storage and Battery Swap.
    8) Custom Home Screens.

3. Architectural diagram of Android operating system

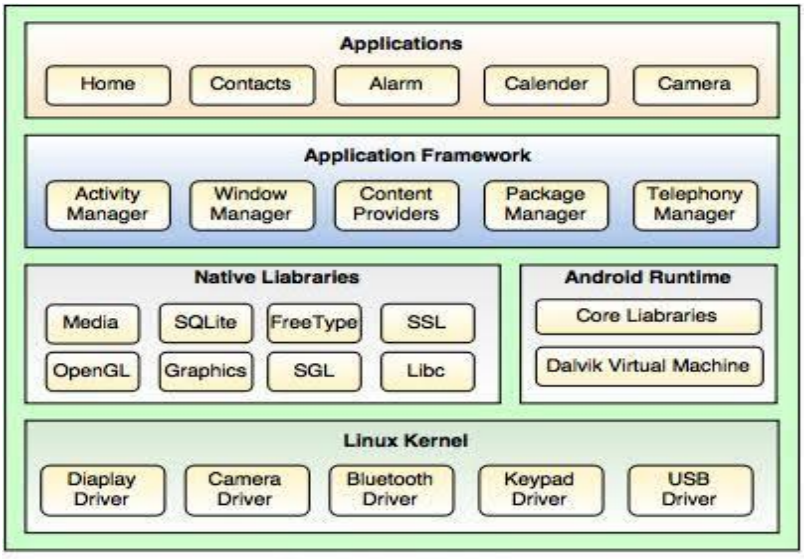

Fig. Android Architecture

4. Differentiate between Windows operating system and Android operating system.

WINDOWS	ANDROID
It is the most used operating system in personal computers.	It is the most used operating system overall.
It is for workstation, personal computers, media center, tablets and embedded systems.	Its target system type is smartphones and tablet computers

EXPERIMENT 2

1)List all the steps to install android operating system

- i)Go to "Android Studio" @ https:// claweloper. android-com | studio ⇒ Click " Download Android Studio 4.1 x for windows cu-bit
- ii) Run the downloaded installer a) In" choose components", Select "Android Studio and "Android Virtual Device (space required: 2-7 GB)
- iii) choose Install Location
- iv)Choose Start Menu Folder -Accept default->Install
- v) Installing will start after completion click on next
- vi) Installation completed click on finish.

2) List various IDEs

- Androis Studio
- Android Web Developer
- Android script
- Python suite
- cppDroid
- Eclipse

3) Explain JVM and DVM

- JVM(Java Virtual Machine)

JVM is the virtual machine that runs java code on different platforms. It acts as an abstract layer between the program and the platform on which the java code is running. The portability of Java code is possible only because of the JVM. The javac compiler converts the source code file(.java file) into an intermediate java bytecode format which is machine/platform independent. This intermediate file is then provided to the target machine/platform where it gets translated into machine code. JVM supports multiple host architecture and it is the reason why Java applications are called WORA(Write Once Run Anywhere).

- DVM(Dalvik Virtual Machine)

DVM is a virtual machine to execute Android applications. The Java bytecode(.class file) generated by javac compiler is converted into Dalvik bytecode to make the application source files executable on the DVM. Since Android devices have a definite processing capacity, memory, and battery life, the DVM design principle aims to optimize itself so that it can load fastly and run smoothly even on low memory/powered devices. This virtual machine is very efficient in running multiple instances on the same device.

4)What is an IDE?

An IDE allows developers to start programming new applications quickly because multiple utilities don't need to be manually configured and integrated as part of the setup process.

EXPERIMENT 3

1)List basic requirements for configuring android operating system?

- A 64-bit environment is required for Android 2.3. x (Gingerbread) and higher versions, including the master branch. ...
- At least 250GB of free disk space to check out the code and an extra 150 GB to build it. ...
- At least 16 GB of available RAM is required, but Google recommends 64 GB.

2)Why bytecode cannot run in Android?

- We cannot run Java Bytecode on Android because: Android uses Dalvik VM(virtual machine) instead of Java VM. To run a Java Bytecode you need JVM(Java Virtual Machine). Java in computers and Android uses a separate environment to run their code.

3)What is a Build Type in Gradle?

- A build type determines how an app is packaged. By default, the Android plug-in for Gradle supports two different types of builds: debug and release .

4)Explain the build process in Android.

- The Android build system compiles app resources and source code, and packages them into APKs that you can test, deploy, sign, and distribute. Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations.

EXPERIMENT 4

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

- Java
- Manifests
- Drawable
- Values
- Build.gradle
- Activity_main.xml

2) What is an activity in Android programming?

- An activity represents a single screen with a user interface just like window or frame of Java.Android activity is the subclass of ContextThemeWrapper class.
- It is program with in an Activity starting with a call on onCreate() callback method.

3) Write a program to display HelloWorld.
4) Write a program to display student name and marks.

- .xml file

```
<?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"
   android:gravity="center"
   tools:context=".MainActivity">
   <TextView
      android:id="@+id/t1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerInParent="true"
      android:layout_centerHorizontal="true"
      android:text="Hello World!"
app:layout_constraintBottom_toBottomOf=
"parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
      android:id="@+id/t2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@id/t1"
      android:layout_centerInParent="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="27dp"
      android:text="Name: ABC"
       />
   <TextView
      android:id="@+id/t3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@id/t2"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="58dp"
      android:gravity="center"
      android:text="marks :88" />
</RelativeLayout>
```

- .java file

```
Package com.example.experiment4;

Import
androidx.appcompat.app.AppCompatA
ctivity;

Import android.os.Bundle;

Import android.widget.TextView;

Public class MainActivity extends
AppCompatActivity {

TextView t1,t2,t3;

    @Override

Protected void onCreate(Bundle
savedInstanceState)

{super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main)
;

  }            }
```

EXPERIMENT 5

1)Name any three-layout manager?

- LinearLayout
- FrameLayout
- RelativeLayout

2) What is Card View?

- 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.

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

- .xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android=
"http://schemas.android.com/apk/res/android"
xmlns:app=
"http://schemas.android.com/apk/res-auto"
xmlns:tools=
"http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity">
   <TextView
      android:id="@+id/t1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Name: ABC" />
   <TextView
      android:id="@+id/t2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Age: 19" />
   <TextView
      android:id="@+id/t3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Mobile no: 1234567890" />
</LinearLayout>
```

- .java file

```
package com.example.exp5;
import
androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends
AppCompatActivity {
TextView t1,t2,t3;
   @Override
   protected void onCreate(Bundle
savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);    }
         }
```

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

- .xml file

```
<?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"
   tools:ignore="Deprecated">
   <TextView
      android:id="@+id/t1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="120dp"
      android:layout_y="230dp"
      android:text="Name:ABC" />
   <TextView
      android:id="@+id/t2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="120dp"
      android:layout_y="280dp"
      android:text="Age: 19"
      />
  <TextView
      android:id="@+id/t3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="120dp"
      android:layout_y="330dp"
      android:text="Phone no: 1234567890" />
```

- .java file

```
package com.example.exp5b;
import
androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends
AppCompatActivity {
TextView t1,t2,t3;
   @Override
   protected void onCreate(Bundle
savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
}
```

EXPERIMENT 6

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

- 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.

2)What is Grid Layout?

- GridLayout uses a grid of infinitely-thin lines to separate its drawing area into: rows, columns, and cells. It supports both row and column spanning, which together allow a widget to occupy a rectangular range of cells that are next to each other.

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

- .xml file

```
<?xml version="1.0"
encoding="utf-8"?>
<TableLayout xmlns:android=
"http://schemas.android.comapk/res/android"
    xmlns:app=
"http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
   tools:context=".MainActivity">
<TableRow>
   <TextView
      android:id="@+id/t1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Name" />
   <TextView
      android:id="@+id/t2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Roll No" />
   <TextView
      android:id="@+id/t3"
      android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="ABC" />
   <TextView
android:id="@+id/t4"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="101" />
   </TableRow>
   <TableRow>
      <TextView
         android:id="@+id/t5"
android:layout_width="wrap_content"
         android:layout_height="wrap_content"
    android:text="PQR" />
       <TextView
          android:id="@+id/t6"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="102" />
     </TableRow>
   </TableRow>
```

```
  <TextView
android:id="@+id/t7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="XYZ" />
<TextView
 android:id="@+id/t8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
 android:text="103" />
    <TableRow>
<TableRow>
 <TextView
    android:id="@+id/t9"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="NMK" />
        <TextView
           android:id="@+id/t10"
     android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="104" />
    </TableRow>
    <TableRow>
        <TextView
           android:id="@+id/t11"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="XY" />
        <TextView
           android:id="@+id/t12"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="105" />
    </TableRow>
    <TableRow>
```

```
        <TextView
           android:id="@+id/t17"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="RTY" />
        <TextView
android:id="@+id/t18"
android:layout_width="wrap_content"
 android:layout_height="wrap_content"
           android:text="106" />
    </TableRow>
    <TableRow>
<TextView
           android:id="@+id/te7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="XY" />
        <TextView
           android:id="@+id/te8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="107" />
    </TableRow>
    <TableRow>
        <TextView
           android:id="@+id/ts7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="YZ" />
        <TextView
           android:id="@+id/ts8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
           android:text="108" />
    </TableRow>
    <TableRow>
```

```
        <TextView
            android:id="@+id/e7"
            android:layout_width="wrap_content"
           android:layout_height="wrap_content"
            android:text="XZ" />
        <TextView
            android:id="@+id/e8"
            android:layout_width="wrap_content"
           android:layout_height="wrap_content"
            android:text="109" />
    </TableRow>
    <TableRow>
        <TextView
            android:id="@+id/e"
            android:layout_width="wrap_content"
android:layout_height="wrap_content"
            android:text="UIK" />
        <TextView
      android:id="@+id/e2"
   android:layout_width="wrap_content"
android:layout_height="wrap_content"
            android:text="110" />
    </TableRow>
</TableLayout>
```

- .java file

```
package com.example.exp5
import
androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends
        AppCompatActivity {
TextView
        t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,ts7,ts8
        ,te7,te8,e,e2,t17,t18,e7,e8;
    @Override
    protected void onCreate(Bundle
        savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
         }        }
```

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

.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android=
"http://schemas.android.com/apk/res/andr
        oid"
xmlns:app="http://schemas.android.com/a
        pk/res-auto"
xmlns:tools="http://schemas.android.com/t
        ools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<Button
android:id="@+id/b1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="display"
android:gravity="center_horizontal"
android:text="Display Data types in OOP"
 />
<TextView
android:id="@+id/ 1"
android:layout_width= "wrap_content"
android:layout_height= "wrap_content"
android:layout_marginTop= "20dp" />
</FrameLayout>
```

.java file:

```
package com.example.exp5;
import
      androidx.appcompat.app.AppCompat
      Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends
      AppCompatActivity {
Button b1;
TextView t1;
   @Override
   protected void onCreate(Bundle
      savedInstanceState) {
        super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main
      );
   t1=(TextView)findViewById(R.id.t1);
   b1=(Button)findViewById(R.id.b1);    }
   public void display(View view)
   {
      t1.setText("\n\nInteger\nCharacter\nB
      oolean\nDouble\nFloat")      }  }
```

EXPERIMENT 7

1X 1. A) Non- visible
IX 2. D) gen

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

- .xml file

```
<?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"
android:orientation="vertical"
tools:context=".MainActivity">
<EditText
      android:id="@+id/et"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Username"  />
  <EditText
      android:id="@+id/et2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Password"
      android:layout_below="@+id/et"     />
<Button
 android:id="@+id/b1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

- .java file

```
package com.example.exp5;
import
androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends
AppCompatActivity
{
Button b1;
EditText et,et2;
TextView tv;
   @Override
   protected void onCreate(Bundle
savedInstanceState)
 {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
et=(EditText)findViewById(R.id.et);
et2=(EditText)findViewById(R.id.et2);
b1=(Button)findViewById(R.id.b1);
tv=(TextView)findViewById(R.id.tv);
 }
   public void display(View view)
   {   String a=et.getText().toString();
```

android:onClick="display"
android:gravity="center_horizontal"
android:text="Login"
android:layout_below="@+id/et2" />
<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/tv"
android:layout_below="@+id/b1"/> </RelativeLayout>

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

- .xml file

```
<?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"
   android:orientation="vertical"
   tools:context=".MainActivity">
   <EditText
      android:id="@+id/et"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Roll No"
      />
   <EditText
      android:id="@+id/et2"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Name"
      android:layout_below="@+id/et"
      />
   <EditText
      android:id="@+id/et3"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Class"
      android:layout_below="@+id/et2"
      />
   <Button
     android:id="@+id/b1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:onClick="display"
     android:gravity="center_horizontal"
```

- .java file

```
package com.example.exp5;
import
androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends
AppCompatActivity {
Button b1;
EditText et,et2,et3;
TextView tv;
   @Override
   protected void onCreate(Bundle
savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
   et=(EditText)findViewById(R.id.et);
       et2=(EditText)findViewById(R.id.et2);
       et3=(EditText)findViewById(R.id.et3);
       b1=(Button)findViewById(R.id.b1);
       tv=(TextView)findViewById(R.id.tv);
   }
   public void display(View view)
   {
     tv.setText("Roll No :"+et.getText().toString()+
           "\nName :"+et2.getText().toString()+
           "\nClass :"+et3.getText().toString()
           );

   }
```

EXPERIMENT 8

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

- Defines the hint displayed in the drop down menu.

2)How to create AutoComplete TextView field in XML?

- In android, by using AutoCompleteTextView control we can create an AutoComplete TextFiels in XML.

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

.XML file

```
<Relative Layout xmlns:
android=http://schemas.android.
com/apk/res/android

-xmlns: tool's=http://schemas.android.com/tools
android:layout_width="match_parent"

Android:layout_height="match_parent" tools:
context a ".MainActivity">

<TextView

Android:id="@+id/t"
Android:layout_width="match_parent"
Android:layout_height="map content"
Android: text="GOOGLE" enduachua

Android: text size = "Podp""

Android gravity="center"/>

<Auto Complete TextViery

Android:id="@+id/edit Text"

Android:layout_width="match_parent"
android:layout_height = "wrap_content"
Android:layout belex = "@+id/t"
Android: hit "Search"/>
</RelativeLayout>
```

.java file

```
Package com.example.exp8-1;

Import android.graphics. Color: import android.os.
Bundlei.

Import android.widget. * ;

Public class MainActivity extends AppCompat
Activity & protected void onCreate(Bundle saved
instanceState) {

@Override

Super.onCreate Csaved InstanceState);

setContent View (R.layout. activity_main);

Bray Adapter <<String> adapter = new Array
Adapter <string> (this.android.R.Layout. select
dialog item, language);

AutoComplete TextView actv = CAutoComplete
Text. Kiew) find
ViewById(R.id.edit Text);
Actv.setThreshold (1);
Actv. Set Adapter (adapter);
Actv. Set Text Color (Color. RED);}
String [] language = {"Facebook.com, "Google.com",
"Yahoo.com"};
}
```

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

.XML file

```
<Relative Layout
xmlns:android=http://scheamas..
android.com/apk/res.landanid.....
android:layout_height match_parent
tools.icantext MainActivity”>

Xmlns:toals
http://schemas.android.com/teels..
android:layout_width=match parent...

<AutoComplete TextView

Android:id=”@+id/editText”

Android:layout_width=”match_parent”
android:layout height=”kap-content”
android:layout belowe=”@+id/t”
android: hint = “Search” 12
/>
</RelativeLayout>
```

.java file

```
Package com.example.exp82;

Import android.app.compat.app.AppCompatfctivity i

Import android.graphics. *; import android.os. Bundle;

Import android.widget. *;

Public class MainActivity extends AppCompatActivity
{( String[] language= {“MAQ”,”WEB”, “PHP”, “ET”
“EDP”,”MGT”}

@Override

Protected veid onCreate CBundle
gavedanstanceState)
{
setContentView(R.layout. activity_main);
ArrayAdapter <<String> adapter = new Anny Adapter
<string> Cthis, android. R.layout. select-dialog item,
language;
AutoComplete Text View actv = findViewById(R.id.

Edit.Texti

Actv...setThreshold C12:

Actv.setAdapter Cadapter)
actv.setTextColox.CColox.RED.);
}}
```

EXPERIMENT 9

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

```
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/button_text"
... />
```

2)How to add image to resources file?

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

3)List four Android Toggle Button control attributes.

- android:id
- android:checked
- android:gravity
- android:text
- android:textOn
- android:textOff

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

.XML file

```
<?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”>
   <ToggleButton
      Android:id=”@+id/togell”
Android:layout_width=”wrap_content”
Android:layout_height=”wrap_content”
Android:layout_centerHorizontal=”true”

      Android:text=”Bluetooth” />
   <TextView

      Android:id=”@+id/t1”

Android:layout_width=”match_parent”

 Android:layout_height=”wrap_content”

 Android:layout_below=”@+id/togell”/>

</RelativeLayout>
```

.java file

```
package com.example.exp9_1;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
   TextView textView;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      textView=findViewById(R.id.t1);
      ToggleButton toggle = findViewById(R.id.togell);

toggle.setOnCheckedChangeListener((buttonView, isChecked) -> {
         if (isChecked) {
            textView.setText("Bluetooth is on");
         } else {
            textView.setText("Bluetooth is off");
         }
      });
   }
}
```

2. Write a program to create a simple calculator.

.XML file

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

<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android

   Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width=”match_parent”
   Android:layout_height=”match_parent”
   Android:orientation=”vertical”
   Android:gravity=”center_horizontal”
   Tools:context=”.MainActivity”>

   <EditText

      Android:id=”@+id/e1”
      Android:layout_width=”wrap_content”
      Android:layout_height=”wrap_content”
      Android:ems=”10”
      Android:inputType=”number” />
   <EditText

      Android:id=”@+id/e2”
      Android:layout_width=”wrap_content”
      Android:layout_height=”wrap_content”
      Android:ems=”10”
      Android:inputType=”number”/>

   <TextView
      Android:id=”@+id/r”
      Android:layout_width=”wrap_content”
      Android:layout_height=”wrap_content”
      Android:text=”TextView”/>
   <Button
      Android:layout_width=”wrap_content”
      Android:layout_height=”wrap_content”
      Android:text=”ADD”
      Android:onClick=”add” />

   <Button

      Android:layout_width=”wrap_content”

      Android:layout_height=”wrap_content”

      Android:text=”SUB”

      Android:onClick=”sub”/>


   <Button

      Android:layout_width=”wrap_content”

      Android:layout_height=”wrap_content”

      Android:text=”MUL”

      Android:onClick=”mul”/>


   <Button

      Android:layout_width=”wrap_content”

      Android:layout_height=”wrap_content”

      Android:text=”DIV”

      Android:onClick=”div”/>


</LinearLayout>
```

.java file

```
Package com.example.myapplication;
Import androidx.appcompat.app.AppCompatActivity;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.EditText;
Import android.widget.TextView;
Public class MainActivity extends AppCompatActivity {
EditText e1,e2;
TextView t;
String a,b;
Integer d;
   @Override
   Protected void onCreate(Bundle savedInstanceState) {
      Super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      e1=findViewById(R.id.e1);
      e2=findViewById(R.id.e2);
      t=findViewById(R.id.r);
   }
   Public void add(View view) {
     A= e1.getText().toString();
     B=e2.getText().toString();
     Integer d=(Integer.valueOf(a)+Integer.valueOf(b));
     t.setText(d.toString());
   }
   Public void sub(View view) {
      A= e1.getText().toString();
      B=e2.getText().toString();
      Integer d=(Integer.valueOf(a)-Integer.valueOf(b));
      t.setText(d.toString()); }
   Public void mul(View view) {
      A= e1.getText().toString();
      B=e2.getText().toString();
      Integer d=(Integer.valueOf(a)*Integer.valueOf(b));
      t.setText(d.toString()); }
   Public void div(View view) {
      A= e1.getText().toString();
      B=e2.getText().toString();
      Integer d=(Integer.valueOf(a)/Integer.valueOf(b));
      t.setText(d.toString());    }}
```

EXPERIMENT 10

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

- The AndroidManifest. xml file

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

- TextView
- EditText
- Buttons
- Checkbox
- Progressbar
- Spinners

- Write a program to create a login form for a social networking site.
- Write a program to create a login form for student registration system

Q. Program to implement login form

.XML file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/
apk/res/android"
xmlns:tools="http://schemas.android.com/to
ols"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">
    <EditText
        android:id="@+id/e1"
       android:layout_width="match_parent"
        android:hint="UserName"
android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/e2"
       android:layout_width="match_parent"
        android:inputType="numberPassword"
        android:hint="PASSWORD"
    android:layout_height="wrap_content"/>
    <TextView
        android:id="@+id/t"
       android:layout_width="match_parent"
     android:layout_height="wrap_content"
       android:gravity="center_horizontal"
/>
    <Button
       android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:text="LOGIN"
        android:onClick="login"
android:layout_gravity="center_horizontal"
/>
</LinearLayout>
```

.java file

```
Package com.example.exp10;
Import
androidx.appcompat.app.AppCompatActivity;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.*;
Public class MainActivity extends
AppCompatActivity {
   @Override
   Protected void onCreate(Bundle
savedInstanceState) {
      Super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
   Public void login(View view) {

      EditText e1=findViewById(R.id.e1);

      EditText e2=findViewById(R.id.e2);

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

      If(e1.getText().toString().equals(“Umesh”)
&& e2.getText().toString().equals(“1234”)){

         t.setText(“Login Success”);   }

      Else{

         t.setText(“Please enter valid details”);    }
}}
```

EXPERMENT 11

1. Name the different methods of Checkbox.

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(boolcon status): Changes the state of the Checkbox

2. List different attributes of Checkbox.

Attribute & Description

- android:drawableBottom This is the drawable to be drawn below the text.
- android:drawableRight This is the drawable to be drawn to the right of the text.
- android:editable If set, specifies that this TextView has an input method.
- android:text This is the Text to display.

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

To define the click event handler for a checkbox, add the android onClick attribute to the <CheckBox> element in your XML layout. The value for this attribute must be the name of the method you want to call in response to a click event. The Activity hosting the layout must then implement the corresponding method.

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

.java file

```
Package com.example.exp11;
Import androidx.annotation.RequiresApi;
Import androidx.appcompat.app.AppCompatActivity;
Import android.os.Build;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.*;
Public class MainActivity extends
AppCompatActivity {
   CheckBox a,b,c,d,e;
   @Override
   Protected void onCreate(Bundle
savedInstanceState) {
Super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
a=(CheckBox) findViewById(R.id.a);  b=(CheckBox)
findViewById(R.id.b);
      c=(CheckBox) findViewById(R.id.c);

      d=(CheckBox) findViewById(R.id.d);

      e=(CheckBox) findViewById(R.id.e);

   }
```

```
   Public void status(View view) {

      StringBuilder r=new
StringBuilder();

      If(a.isChecked())
{r.append("CheckBox1 "); }

      If(b.isChecked())
{r.append("CheckBox2 "); }

      If(c.isChecked())
{r.append("CheckBox3 "); }

      If(d.isChecked())
{r.append("CheckBox4 "); }
      If(e.isChecked())
{r.append("CheckBox5 "); }
Toast.makeText(getApplicationContext
(),result.toString(),Toast.LENGTH_SH
ORT).show();

   }
```

.XML file

```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android
 Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width=”match_parent”
   Android:layout_height=”match_parent”
   Android:orientation=”vertical”
   Tools:context=”.MainActivity”>

   <CheckBox
      Android:id=”@+id/a”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
      Android:text=”CheckBox1” />
   <CheckBox
      Android:id=”@+id/b”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
      Android:text=”CheckBox2” />
   <CheckBox
      Android:id=”@+id/c”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
      Android:text=”CheckBox3” />
   <CheckBox
      Android:id=”@+id/d”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
      Android:text=”CheckBox4” />
   <CheckBox
      Android:id=”@+id/e”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
      Android:text=”CheckBox5” />
   <Button
      Android:id=”@+id/button”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content”
         Android:onClick=”status”
      Android:text=”Check Status” />

</LinearLayout>
```

EXPERIMENT 12

1. Write xml tag to create a Radio button.

To define the click event handler for a button, add the android:onClick attribute to the <RadioButton> element in your XML layout.

2. Write the purpose of Radio Button

Radio buttons allow the user to select one option from a set. You should use radio buttons for optional sets that are mutually exclusive if you think that the user needs to see all available options side-by-side. If it's not necessary to show all options side-by-side, use a spinner instead.

3. List different methods of Radio Button

Following are the few methods of radio button

- 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.
- clearCheck() It clears the selection. When the selection is cleared, no radio button in this group is selected and getCheckedRadioButtonld() returns null
- getCheckedRadioButtonId() It returns the identifier of the selected radio button in this group. If its empty selection, the returned value is-1.
- 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.

4. 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

.java file

```
Package com.example.exp_11;
Import
androidx.appcompat.app.AppCompatActivity;
Import android.os.Bundle;
Import android.view.View;
Import android.widget.RadioButton;
Import android.widget.Toast;
Public class MainActivity extends
AppCompatActivity {
   RadioButton r1,r2,r3,r4;
   @Override
   Protected void onCreate(Bundle
savedInstanceState) {
Super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);   }
```

```
Public void show(View view) {
      R1=findViewById(R.id.r1);
      R2=findViewById(R.id.r2);
      R3=findViewById(R.id.r3);
      R4=findViewById(R.id.r4);
      StringBuilder s=new StringBuilder();
      If (r1.isChecked()) {s.append("Radio
Button 1 "); }
      If (r2.isChecked()) {s.append("Radio
Button 2 "); }
      If (r3.isChecked()) {s.append("Male ");}
     If(r4.isChecked()){s.append("Female");}
      Toast.makeText(this,s.toString()+" is
Selected",Toast.LENGTH_SHORT).show();
}}
```

.XML file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android
   Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width="match_parent"
   Android:layout_height="match_parent"
   Android:orientation="vertical"
   Tools:context=".MainActivity">
   <TextView
      Android:layout_width="match_parent"
      Android:layout_height="wrap_content"
      Android:gravity="center_horizontal"
      Android:text="Single Radio Buttons" />
   <RadioButton
      Android:id="@+id/r1"
      Android:layout_width="match_parent"
      Android:layout_height="wrap_content"
      Android:text="Radio Button 1" />
   <RadioButton
      Android:id="@+id/r2"
      Android:layout_width="match_parent"
      Android:layout_height="wrap_content"
      Android:text="Radio Button 2" />
   <TextView
      Android:id="@+id/textView2"
      Android:layout_width="match_parent"
      Android:layout_height="wrap_content"
      Android:layout_marginTop="10dp"
      Android:gravity="center_horizontal"
     Android:text="Radio Button inside
RadioGroup" />
```

RadioGroup

```
Android:layout_width="match_parent"

Android:layout_height="wrap_content"
>

      <RadioButton
        Android:id="@+id/r3"

Android:layout_width="match_parent"

Android:layout_height="wrap_content"

         Android:text="Male" />

      <RadioButton

         Android:id="@+id/r4"

Android:layout_width="match_parent"

Android:layout_height="wrap_content"

         Android:text="Female" />

   </RadioGroup>



   <Button


Android:layout_gravity="center_horizon
```

EXPERIMENT 13

1. State different methods to update the per entage of progress displayed.

- 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.

2. Write an xml tag for the determinate progress bar.

- To add a progress bar to a layout (xml) file, you can use the <Progress Bar> element By default, a progress bar is a spinning wheel (an indeterminate indicator). To change to a horizontal progress bar, apply the progress bar's horizontal style

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

- Indeterminate Progress

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

- Progress Dialog

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.

1. Write a program to display circular progress bar.

.java file

```
Package com.example.exp13;
Import
androidx.appcompat.app.AppCompatA
ctivity;
Import android.os.Bundle;
Import android.widget.ProgressBar;
Public class MainActivity extends
AppCompatActivity {
   ProgressBar p;
   @Override
   Protected void onCreate(Bundle
savedInstanceState) {
Super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main)
;  }}
```

.XML file

```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/andro
id   Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width=”match_parent”
   Android:layout_height=”match_parent”
   Android:gravity=”center”
   Tools:context=”.MainActivity”>
   <ProgressBar
      Android:id=”@+id/progressBar”
      Style=”?android:attr/progressBarStyle”
      Android:layout_width=”wrap_content”
      Android:layout_height=”wrap_content”
      Tools:layout_editor_absoluteX=”73dp”
      Tools:layout_editor_absoluteY=”96dp” />
</LinearLayout>
```

2. Write a program to show the following output.

.XML file

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

<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android

Xmlns:tools=http://schemas.android.com/tools

Android:layout_width=”match_parent”

Android:layout_height=”match_parent”

   Android:orientation=”vertical”

   Android:gravity=”center”

   Tools:context=”.MainActivity”>

   <Button
      Android:layout_width=”100dp”

      Android:layout_height=”50dp”

      Android:onClick=”show”

      Android:text=”Button” />

</LinearLayout>
```

.java file

```
Package com.example.exp13;
Import androidx.appcompat.app.AppCompatActivity;
Import android.app.ProgressDialog;
Import android.os.Bundle;
Import android.view.View;
Public class MainActivity extends AppCompatActivity {
      Private ProgressDialog p;
   @Override
   Protected void onCreate(Bundle savedInstanceState) {
       Super.onCreate(savedInstanceState);

       setContentView(R.layout.activity_main);
}
   Public void show(View view) {
    P=new ProgressDialog(this);

       p.setMessage(“File Downloading...”);
p.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
       p.setProgress(0);
       p.show();
       final int totalProgressTime = 100;
       final Thread t = new Thread() {
          @Override
          Public void run() {
             Int jumpTime = 1;
             While(jumpTime < totalProgressTime) {
                Try {
                    Sleep(200);
                   jumpTime= jumpTime+5;
                    p.setProgress(jumpTime);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }       }          }   };
  t.start();
   }
   }
```

EXPERIMENT 14

1. List all attributes of Image View.

Attribute	Description
android:maxHeight	Used to specify a maximum height for this view.
android:maxWidth	Used to specify a maximum width for this view.
android src	Sets a drawable as the content for this ImageView.
android:scaleType	Controls how the image should be resized or moved to match the the ImageView.

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

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

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

- android:stretchMode
  none − Stretching is disabled.
  spacingWidth − The spacing between each column is stretched.
  columnWidth − Each column is stretched equally.

spacingWidthUniform − The spacing between each column is uniformly stretched..

1. Write a program to show the following output. Use appropriate view for the same.
   14.1

   MainActivity.java

   Package com.example.exp14_1;

   Import androidx.appcompat.app.AppCompatActivity;

   Import android.os.Bundle;

   Import android.view.View;

   Import android.widget.*;
   Public class MainActivity extends AppCompatActivity {

   String[] mobileArray = {“Android”,”Java”,”Php”,”Hadoop”,

   “Sap”,”Python”,”Ajax”,”C++”,”Ruby”,”Rails”,”.Net”};

   @Override

```
    Protected void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ArrayAdapter adapter = new ArrayAdapter<String>(this,
                R.layout.activity_listview, mobileArray);

        ListView listView = (ListView) findViewById(R.id.mobile_list);
        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            Public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {
                String value=adapter.getItem(position).toString();
                Toast.makeText(getApplicationContext(),value,Toast.LENGTH_SHORT).show();

            }
        });
    }
}
```

Activity_main.xml

```
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
   Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width=”match_parent”
   Android:layout_height=”match_parent”
   Android:orientation=”vertical”
   Tools:context=”.MainActivity” >
   <ListView
      Android:id=”@+id/mobile_list”
      Android:layout_width=”match_parent”
      Android:layout_height=”wrap_content” >
   </ListView>
</LinearLayout>
```

Activity_listview.xml

```
<TextView xmlns:android=http://schemas.android.com/apk/res/android
   Android:id=”@+id/label”
   Android:layout_width=”fill_parent”
   Android:layout_height=”fill_parent”
   Android:padding=”10dip”
   Android:textSize=”16dip”
   Android:textStyle=”bold” >
</TextView>
```

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.
14.2

```
Package com.example.exp14_2;

Import androidx.appcompat.app.AppCompatActivity;

Import android.os.Bundle;

Import android.view.View;

Import android.widget.ImageView;


Public class MainActivity extends AppCompatActivity {

ImageView iw;
   @Override
   Protected void onCreate(Bundle savedInstanceState) {
      Super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      iw=findViewById(R.id.imageView);
   }
   Public void change(View view) {
      Iw.setImageResource(R.drawable.ic_launcher_background);
   }
}

<?xml version=”1.0” encoding=”utf-8”?>

<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
   Xmlns:tools=http://schemas.android.com/tools
```

```
Android:layout_width=”match_parent”
Android:layout_height=”match_parent”
Android:orientation=”vertical”
Tools:context=”.MainActivity”>

<ImageButton
    Android:id=”@+id/imageView”
    Android:layout_width=”match_parent”
    Android:layout_height=”300dp”
    Android:src=”@drawable/ic_launcher_foreground” />
<Button
    Android:layout_width=”wrap_content”
    Android:layout_height=”wrap_content”
    Android:text=”Change Image”
    Android:layout_gravity=”center_horizontal”
    Android:onClick=”change”/>

</LinearLayout>
```

3. Write a program to display 15 buttons using grid view.
14.3

MainActivity.java

```
Package com.example.exp14_3;

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 ImageAdapter(this));
    }
}
```

ImageAdapter.java

```
Package com.example.exp14_3;
Import android.content.Context;
Import android.view.*;
Import android.widget.*;

Public class ImageAdapter extends BaseAdapter  {
    Private Context mContext;
    Private int btn_id;
    Private int total_btns = 15;
    Public ImageAdapter(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;
```

```
        }


        Return btn;
    }
}
```

Activity_main.xml

```
<?xml version=”1.0” encoding=”utf-8”?>
<GridView xmlns:android=http://schemas.android.com/apk/res/android
    Xmlns:tools=http://schemas.android.com/tools
    Android:layout_width=”fill_parent”
    Android:layout_height=”wrap_content”
    Android:numColumns=”5”
    Android:id=”@+id/gridview”
    Tools:context=”.MainActivity”>
</GridView>
```

Activity_gridview.xml

```
<?xml version=”1.0” encoding=”utf-8”?>
<RelativeLayout
    Xmlns:android=http://schemas.android.com/apk/res/android
    Android:layout_width=”match_parent”
    Android:layout_height=”match_parent”>
<Button
    Android:layout_width=”wrap_content”
```

```
   Android:layout_height=”wrap_content”/>
</RelativeLayout>
```

4. Write a program to display a text view using vertical scroll view.
   14.4

```
Package com.example.exp14_4;

Import androidx.appcompat.app.AppCompatActivity;

Import android.os.Bundle;

Public class MainActivity extends AppCompatActivity {

   @Override
   Protected void onCreate(Bundle savedInstanceState) {
      Super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
}
<ScrollView xmlns:android=http://schemas.android.com/apk/res/android
   Xmlns:tools=http://schemas.android.com/tools
   Android:layout_width=”match_parent”
   Android:orientation=”vertical”
   Android:layout_height=”match_parent”
   Tools:context=”.MainActivity”>
```

```
    <TextView
        Android:layout_width=”match_parent”
        Android:layout_height=”wrap_content”
        Android:textSize=”140dp”
        Android:text=”Hello This is a Text view eg.” />
</ScrollView>
```

Experiment 15

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?

- Toast Positioning
  You can change the positioning on the screen of a Toast message using the

setGravity() method. Here is a Toast setGravity()

2. List two constants of Toastclass.

- The Toast class contains two predefined constants you can use: Toast. LENGTH_SHORT and Toast. LENGTH_LONG .

1. Write a program to display following toast message.
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.

PROGRAM :

```
Package com.example.exp15;
Import androidx.appcompat.app.AppCompatActivity;

Import android.os.Bundle;

Import android.view.Gravity;

Import android.view.LayoutInflater;

Import android.view.View;

Import android.view.ViewGroup;

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 view) {
        LayoutInflater li = getLayoutInflater();
        View layout = li.inflate(R.layout.custom,(ViewGroup) findViewById(R.id.custom));
        Toast toast = new Toast(getApplicationContext());
        Toast.setDuration(Toast.LENGTH_SHORT);
        Toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
        Toast.setView(layout);
        Toast.show();
    }
}

<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
    Xmlns:tools=http://schemas.android.com/tools
    Android:layout_width=”match_parent”
    Android:orientation=”vertical”
    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=”Hello World Toast Example” />

    <Button
        Android:id=”@+id/button”
        Android:layout_width=”wrap_content”
        Android:layout_height=”wrap_content”
        Android:onClick=”show”
        Android:text=”Show Toast” />
</LinearLayout>
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout
    Xmlns:android=http://schemas.android.com/apk/res/android
    Android:layout_width=”match_parent”
    Android:orientation=”vertical”
    Android:background=”@color/black”
    Android:layout_height=”wrap_content”>

    <TextView
        Android:layout_width=”match_parent”
        Android:layout_height=”wrap_content”
        Android:textSize=”35dp”
        Android:textColor=”@color/white”
        Android:text=”Message for you:”
 />
```

```
    <TextView
        Android:layout_width=”match_parent”
        Android:layout_height=”wrap_content”
        Android:textColor=”@color/white”
        Android:text=”You have got mail.”
        Android:textSize=”28dp”
        />
</LinearLayout>
```

```
Package com.example.exp_152;

Import androidx.appcompat.app.AppCompatActivity;

Import android.os.Bundle;
Import android.view.View;
Import android.widget.*;

Public class MainActivity extends AppCompatActivity {
    CheckBox c1,c2,c3;
    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
    }

    Public void Order(View view) {
        C1=findViewById(R.id.c1);
        C2=findViewById(R.id.c2);
        C3=findViewById(R.id.c3);

        StringBuilder s= new StringBuilder();
        Int t=0;

        If(c1.isChecked()){s.append(“Coffe 50Rs”); t+=50;}
        If(c2.isChecked()){s.append(“\nBurger 120Rs”); t+=120;}
        If(c3.isChecked()){s.append(“\nPizza 170Rs”); t+=170;}

        s.append(“\nTotal: “+t+”RS”);
        Toast.makeText(this,s.toString(), Toast.LENGTH_SHORT).show();
    }
}
```

```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout xmlns:android=http://schemas.android.com/apk/res/android
    Xmlns:tools=http://schemas.android.com/tools
    Android:layout_width=”match_parent”
    Android:layout_height=”match_parent”
    Android:orientation=”vertical”
    Android:gravity=”center_horizontal”
    Tools:context=”.MainActivity”>

    <CheckBox
        Android:id=”@+id/c1”
        Android:layout_width=”100dp”
        Android:layout_height=”wrap_content”
        Android:text=”Pizza” />

    <CheckBox
        Android:id=”@+id/c2”
        Android:layout_width=”100dp”
        Android:layout_height=”wrap_content”
        Android:text=”Coffee” />
```

```
    <CheckBox
        Android:id=”@+id/c3”
        Android:layout_width=”100dp”
        Android:layout_height=”wrap_content”
        Android:text=”Burger” />

    <Button
        Android:layout_width=”wrap_content”
        Android:layout_height=”wrap_content”
        Android:layout_marginTop=”250dp”
        Android:onClick=”Order”
        Android:text=”ORDER” />

</LinearLayout>
```

Experiment 16

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

- Time Picker:

Android Time Picker allows you to select the time of day in either 24 hour or AMPM 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

```
package example.javatpoint.com.timepicker;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;
```

2. List and explain all methods of TimePickerclass

- In order to get the time selected by the user on the screen, you will use getCurrentHour() and getCurrentMinute() method of the TimePicker Class. Their syntax is given below.

```
int hour = timePicker1.getCurrentHour();
int min = timePicker1.getCurrentMinute();
```

3. List and explain any five methods of DatePickerclass.

- Method & description

1 getDayOfMonth() This method gets the selected day of month

2 getMonth() This method gets the selected month

3 getYear() This method gets the selected year

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

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

PROGRAM :

16.1

```
Package com.example.exp16;


Import androidx.appcompat.app.AppCompatActivity;

Import android.os.Bundle;
```

```
Import android.view.View;
Import android.widget.TimePicker;

Public class MainActivity extends AppCompatActivity {
    TimePicker t;
    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t=findViewById(R.id.timePicker1);
        t.setIs24HourView(true);
    }
    }

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

```
    Android:orientation=”vertical”>
    <TimePicker
        Android:layout_width=”match_parent”
        Android:layout_height=”200dp”
        Android:timePickerMode=”spinner”
       />

    <TimePicker
        Android:layout_width=”match_parent”
        Android:layout_height=”wrap_content”
        Android:timePickerMode=”spinner”
        Android:id=”@+id/timePicker1”  />

    <TimePicker
        Android:layout_width=”wrap_content”
        Android:layout_height=”wrap_content”
        Android:timePickerMode=”clock”/>
</LinearLayout>
</ScrollView>
```

16.2

```
Package com.example.exp16_2;

Import android.app.*;
```

```
Import android.os.Bundle;
Import android.view.View;
Import android.widget.*;

Import androidx.appcompat.app.AppCompatActivity;

Import java.util.Calendar;

Public class MainActivity extends AppCompatActivity implements
        View.OnClickListener {

    Button btnDatePicker, btnTimePicker;
    EditText txtDate, txtTime;
    Private int mYear, mMonth, mDay, mHour, mMinute;

    @Override
    Protected void onCreate(Bundle savedInstanceState) {
        Super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnDatePicker=(Button)findViewById(R.id.btn_date);
        btnTimePicker=(Button)findViewById(R.id.btn_time);
        txtDate=(EditText)findViewById(R.id.in_date);
        txtTime=(EditText)findViewById(R.id.in_time);
```

```
    btnDatePicker.setOnClickListener(this);
    btnTimePicker.setOnClickListener(this);

}

@Override
Public void onClick(View v) {

    If (v == btnDatePicker) {


        Final Calendar c = Calendar.getInstance();
        mYear = c.get(Calendar.YEAR);
        mMonth = c.get(Calendar.MONTH);
        mDay = 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) {

                        txtDate.setText(dayOfMonth + “-“ + (monthOfYear + 1) + “-“ + year);
```

```
                }
            }, mYear, mMonth, mDay);
        datePickerDialog.show();
    }
    If (v == btnTimePicker) {

        Final Calendar c = Calendar.getInstance();
        mHour = c.get(Calendar.HOUR_OF_DAY);
        mMinute = c.get(Calendar.MINUTE);

        TimePickerDialog timePickerDialog = new TimePickerDialog(this,
                New TimePickerDialog.OnTimeSetListener() {

                    @Override
                    Public void onTimeSet(TimePicker view, int hourOfDay,
                                    Int minute) {

                        txtTime.setText(hourOfDay + ":" + minute);
                    }
            }, mHour, mMinute, false);
        timePickerDialog.show();
    }
  }
}
```

```
<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”>

    <EditText
        Android:layout_width=”200dp”
        Android:layout_height=”wrap_content”
        Android:id=”@+id/in_date”
        Android:layout_marginTop=”82dp”
        Android:layout_alignParentTop=”true”
        Android:layout_alignParentLeft=”true”
        Android:layout_alignParentStart=”true” />

    <Button
        Android:layout_width=”wrap_content”
        Android:layout_height=”wrap_content”
        Android:text=”SELECT DATE”
        Android:id=”@+id/btn_date”
        Android:layout_alignBottom=”@+id/in_date”
        Android:layout_toRightOf=”@+id/in_date”
        Android:layout_toEndOf=”@+id/in_date” />
```

```
    <EditText
        Android:layout_width=”200dp”
        Android:layout_height=”wrap_content”
        Android:id=”@+id/in_time”
        Android:layout_below=”@+id/in_date”
        Android:layout_alignParentLeft=”true”
        Android:layout_alignParentStart=”true” />

    <Button
        Android:layout_width=”wrap_content”
        Android:layout_height=”wrap_content”
        Android:text=”SELECT TIME”
        Android:id=”@+id/btn_time”
        Android:layout_below=”@+id/btn_date”
        Android:layout_alignLeft=”@+id/btn_date”
        Android:layout_alignStart=”@+id/btn_date” />

</RelativeLayout>
```