

Experiment No 1

Date :

Problem Definition :

Develop an application that uses GUI components, Fonts and Colors.

Objectives

1. To develop a simple android application that uses GUI components, Font and color.
2. To develop an Android application in Android Studio
3. To understand the XML components working used in the design of an application

Theory

functions used in the program.

1) onCreate()

The `onCreate()` method is where any initialization code should go, as this method always gets called after the activity has launched but before it starts running.

2) setContentView()

It sets the XML file you want as your main layout when the app starts.

3) findViewById()

This method is used to find an existing view in your XML layout by its `android:id` attribute.

4) setOnClickListener()

This method links a listener with certain attributes. It is basically used with buttons, image buttons etc. While invoking this method a callback function will run.

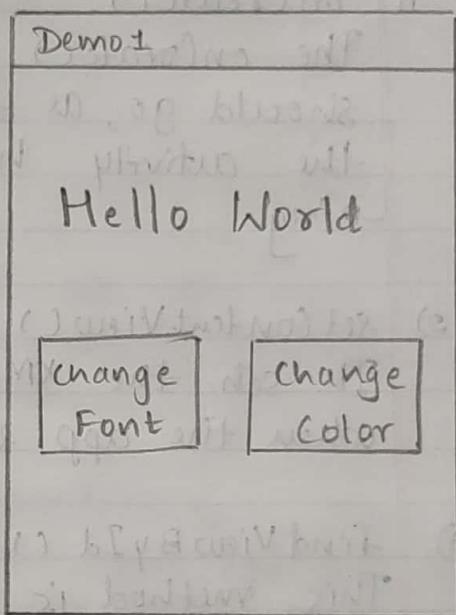
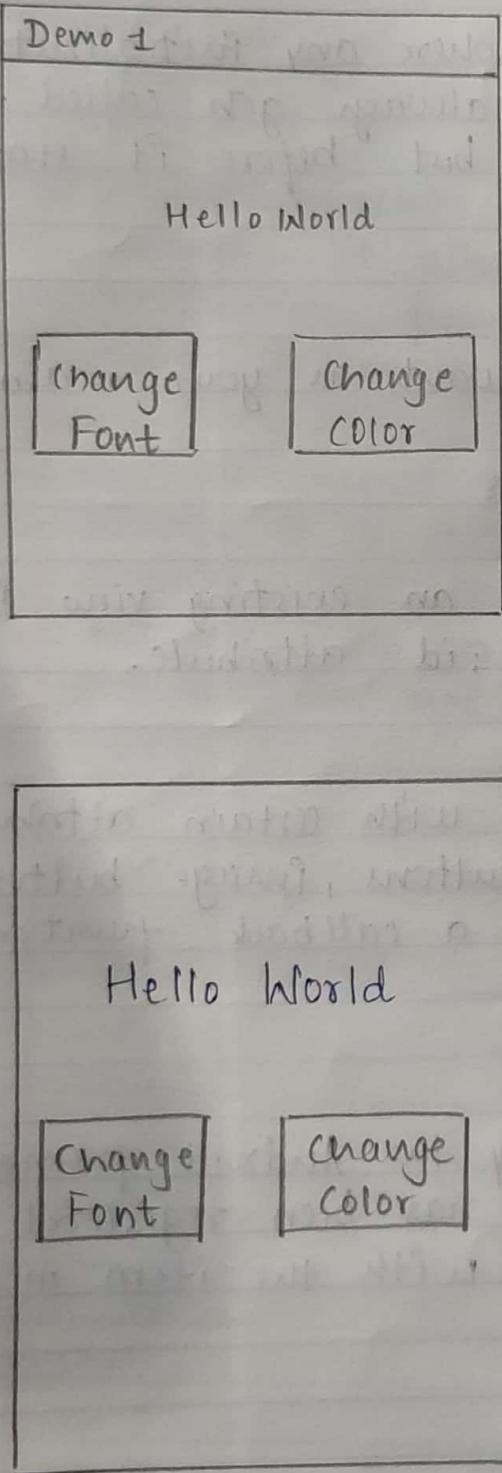
5) onClick()

This method will be called by the Android framework to see that the listener that has been registered is triggered by user interaction with the item in the UI.

6) setTextColor()

sets the text color.

Diagram.



Descriptions of packages used.

1. `java.util.Random` : This package contains classes and interfaces that support a generic API for random number generation.
2. `android.graphics.color` : Color is a theming library for Android. It allows our app to use over 4,000 color combinations from the material design palette by separately picking a primary and color.
3. `android.support.v7.app.AppCompatActivity` : The support library is a static library that you can add to our android application in order to use API's that are either not available for older platform versions or utility API that are not a part of the framework APIs.

Source Code.

Activity.xml

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

< TextView

```
    android:id = "@+id/textView"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:layout_margin = "30dp"
    android:gravity = "center"
    android:text = "Hello World!"
    android:textSize = "25sp"
    android:textStyle = "bold" />
```

< Button

```
    android:id = "@+id/button1"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:layout_margin = "20dp"
    android:gravity = "center"
    android:text = "Change font size"
    android:textSize = "25sp" />
```

< Button

```
    android:id = "@+id/button2"
    android:layout_width = "match_parent"
```

```
    android:layout_height = "wrap_content"  
    android:layout_margin = "20dp"  
    android:gravity = "center"  
    android:text = "change color"  
    android:textSize = "25sp" />  
</LinearLayout>
```

MainActivity.java

```
package com.example.cmn1;  
import android.graphics.Color;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity  
{  
    int ch;  
    float fout = 30;  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        final TextView t = (TextView) findViewById(R.id.textView);  
        Button b1 = (Button) findViewById(R.id.button1);  
        b1.setOnClickListener(new View.OnClickListener() {  
            @Override
```

```
public void onClick (View v) {  
    t.setTextSize (fout);  
    fout = fout + 5;  
    if (fout == 50)  
        fout = 30;  
}  
});
```

```
Button b2 = (Button) findViewById (R.id.button2);  
b2.setOnClickListener (new View.OnClickListener () {
```

@Override

```
public void onClick (View v) {  
    switch (ch) {
```

case 1:

```
    t.setTextColor (Color.RED);  
    break;
```

case 2:

```
    t.setTextColor (Color.GREEN);  
    break;
```

case 3:

```
    t.setTextColor (Color.BLUE);  
    break;
```

case 4:

```
    t.setTextColor (Color.CYAN);  
    break;
```

}

```
    ch++;  
    if (ch == 7)  
        ch = 1;
```

}

} ;

}

Expected Output

- A android application consisting of a text and two buttons in its interface , in which the first button is responsible for the increase in the text size and the second button is responsible for the change in the above text color.
- Initial Text Size is 30 , once change font button is pressed there should be increase in the text size by 5.
- Once the change color button is pressed , the color of the text must be changed that is either Red, Green, Blue, Cyan, Yellow etc.

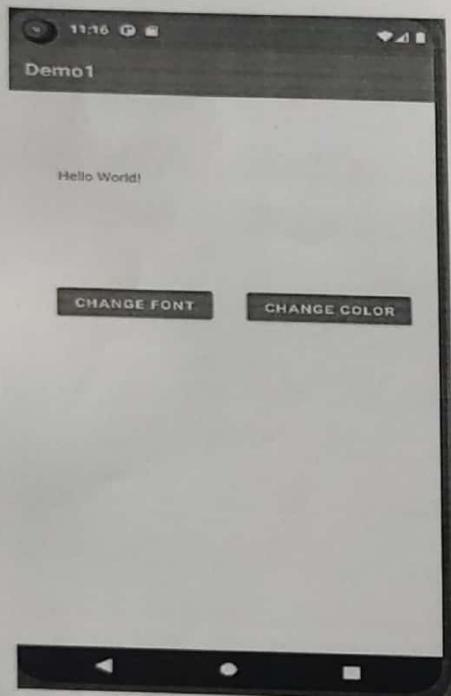
TERMWORK 1

DATE: 31/10/2022

USN: 2GI19CS077

NAME: Namrata S Shivakale

OUTPUT:



Namrata
14/11/22

Conclusion.

The application runs as expected , the text size and text color can be changed on the respective Buttons

References:

Android Studio 3.5 Development essentials : Java Edition , 2019 Neil Smyth / Payload Media , Inc

Experiment No 2.

Date :

Problem Definition

Develop an application that uses layout managers and event listeners

Objectives.

1. To understand the use of layout managers
2. To understand the use of event listeners
3. To understand how to pass data in case of multiple layouts
4. To develop a simple application using layout managers and event listeners.

Theory

functions used in the program

1. onCreate()

The onCreate() method is where any initialization code should go, as this method always gets called after the activity has launched but before it starts running.

2. setContentView()

It sets the XML file you want as your main layout when the app starts

3. findViewById()

This method finds an existing view in your XML layout by its android:id attribute

4. setOnClickListener()

This method links the listener with certain attributes. While invoking this method a callback function will run

5. onClick()

This method is used to see that the listener that has been registered is triggered by user interaction with the item in the UI.

6. getText()

This method is used to get Text from controls like EditText and Button.

7. putExtra()

This method is used to add extended data to the Intent.

Diagram

TW2	
Details Form	
Name	<u>Nomata</u>
Reg. No.	<u>77</u>
Dept	<u>CSE</u>

TW2	
Details Form	
Name	<u>Nomata</u>
Reg. No.	<u>77</u>
Dept	<u>CSE</u>

8. `getStringExtra()`

This method is used for getting the data (key) that is sent by the above method. `putExtra()` method.

9. `setText()`

This method sets the text to be displayed using a string resource identifier.

Layouts used in this application are as follows.

1. `LinearLayout` → It aligns all children in a single direction, vertically or horizontally.
2. `RelativeLayout` → It displays child views in relative position.
3. `Grid Layout` → It displays items in 2-D manner.

Source Code.

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" >
```

<LinearLayout

```
    android:layout_width="match-parent"
    android:layout_height="100dp" >
```

< TextView

```
    android:id="@+id/TextView"
    android:layout_width="match-parent"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="Details Form"
    android:textSize="25sp"
    android:gravity="center" />
</LinearLayout >
```

< GridLayout

```
    android:id="@+id/gridLayout"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    android:layout_marginTop="100dp"
    android:layout_marginBottom="200dp"
    android:columnCount="2"
```

```
    android:rowCount = "3" >
```

```
< TextView
```

```
    android:id = "@+id/textView1"
```

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

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

```
    android:layout_margin = "10dp"
```

```
    android:layout_row = "0"
```

```
    android:layout_column = "0"
```

```
    android:text = "Name"
```

```
    android:textSize = "20sp"
```

```
    android:gravity = "center" />
```

```
< EditText
```

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

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

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

```
    android:layout_margin = "10dp"
```

```
    android:layout_row = "0"
```

```
    android:layout_column = "1"
```

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

```
< TextView
```

```
    android:id = "@+id/textView2"
```

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

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

```
    android:layout_margin = "10dp"
```

```
    android:layout_row = "1".
```

```
    android:layout_column = "0"
```

```
    android:text = "Reg No."
```

```
    android:textSize = "20sp"
```

```
        android:gravity = "center" />
```

<EditText

```
    android:id = "@+id/editText2"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_margin = "10dp"
    android:layout_row = "1"
    android:layout_column = "1"
    android:inputType = "number"
    android:ems = "10" />
```

<TextView

```
    android:id = "@+id/textView03"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_margin = "10dp"
    android:layout_row = "2"
    android:layout_column = "0"
    android:text = "Dept"
    android:textSize = "20sp"
    android:gravity = "center" />
```

<Spinner

```
    android:id = "@+id/spinner"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_margin = "10dp"
    android:layout_row = "2"
    android:layout_column = "1"
    android:spinnerMode = "dropdown" />
```

<GridLayout >

<Button

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="15dp"
    android:text="Submit" />
```

</RelativeLayout >

Activity Second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context="com.example.devang.exno2.SecondActivity"
    android:orientation="vertical"
    android:gravity="center" />
```

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="New Text"
    android:textSize="30sp" />
```

< TextView

```
    android:id = "@+id/TextView2"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_margin = "20dp"
    android:text = "New Text"
    android:textSize = "20sp" />
```

< TextView

```
    android:id = "@+id/TextView3"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_margin = "20dp"
    android:text = "New Text"
    android:textSize = "20sp" />
</LinearLayout >
```

MainActivity.java

```
package com.example.exnoe;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    // Defining the Views
```

```
    EditText e1, e2;
```

```
    Button bt;
```

```
    Spinner s;
```

```
    // Data for populating in spinner
```

```
    String [] dept_array = {"CSE", "EEE", "IT", "Mech", "Civil"};
```

```
    String name, reg, dept;
```

```
@Override
```

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

```
    super.onCreate(savedInstanceState);
```

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

```
    // Referring the Views
```

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

```
    e2 = (EditText) findViewById(R.id.editText2);
```

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

```
    s = (Spinner) findViewById(R.id.spinner);
```

```
    // creating adapter for spinner for adapting the data  
    // from array to spinner
```

```
    ArrayAdapter adapter = new ArrayAdapter(MainActivity.this,  
        android.R.layout.simple_spinner_item, dept_array);
```

```
    s.setAdapter(adapter);
```

```
    // creating listener for Button
```

```
    bt.setOnClickListener(new View.OnClickListener()
```

@Override

public void onClick (View v)

// Getting the values from views

name = e1.getText().toString();

reg = e2.getText().toString();

dept = s.getSelectedItem().toString();

Intent i = new Intent (MainActivity.this, SecondActivity.class);

i.putExtra ("name-key", name);

i.putExtra ("reg-key", reg);

i.putExtra ("dept-key", dept);

startActivity (i);

}

});

g

SecondActivity.java

package com.example.exno2;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

TextView t1, t2, t3;

String name, reg, dept;

@Override

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

t1 = (TextView) findViewById(R.id.textView1);

t2 = (TextView) findViewById(R.id.textView2);

t3 = (TextView) findViewById(R.id.textView3);

Intent i = getIntent();

// Getting the values from First Activity using the Intent Received

name = i.getStringExtra("name-key");

reg = i.getStringExtra("reg-key");

dept = i.getStringExtra("dept-key");

// Setting the Values to Intent

t1.setText(name);

t2.setText(reg);

t3.setText(dept);

5

4

Expected Output

- A android application that take student's details such as Name, Reg .No and Department. submit button is has to be created to press once the information is typed.
- Once all the student details are typed and submit button is pressed , all the details has to be displayed in the separate Activity.

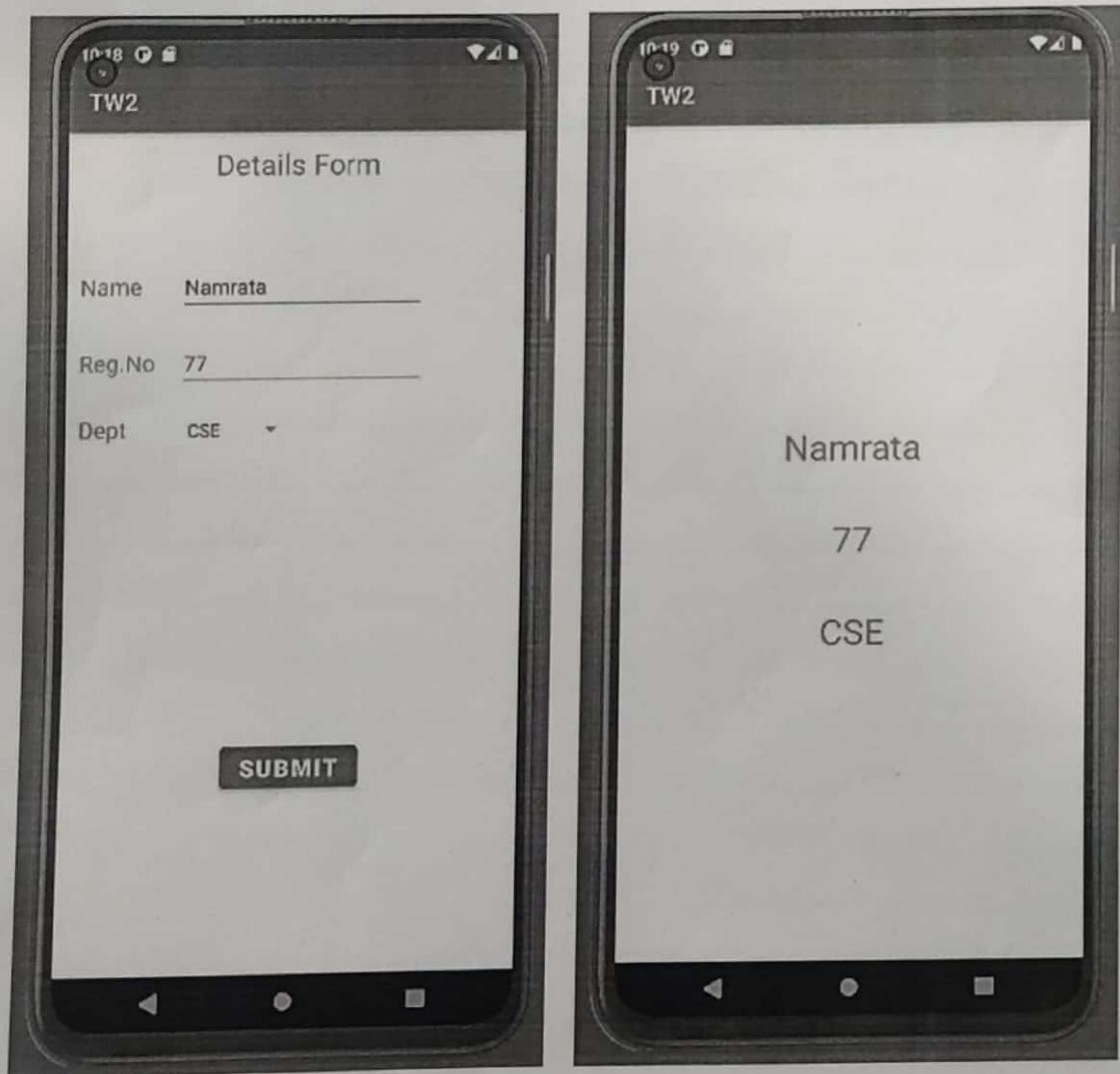
TERMWORK – 2

Date : 14-11-2022

Name : Namrata Shivakale

USN : 2GI19CS077

OUTPUT:



*Namrata
14/11/22*

Conclusion

By this experiment, we understood basics of layout managers, event listeners, intent. we also understood how to pass data from one activity to another and implemented the same.

References:

Android Studio 3.5 Development Essentials, Java Edition,
2019 Neil Smyth / Payload Media, Inc.

Experiment No 3

Date:

Problem Definition

Develop a native calculator Application.

Objectives :

1. To understand the use of LinearLayout
2. To understand the use of event listeners
3. To understand the use of components
4. To develop a calculator application.

Theory

functions used in the programs are as follows.

1. onCreate()

The `onCreate()` method is where any initialization code should go, as this method always gets called after the activity has launched but before it starts running.

2. setContentView()

It sets the XML file you want as your main layout when the app starts.

3. findViewById()

This method finds an existing view in your XML layout by its `android:id` attribute.

4. setOnClickListener()

This method links the listener with certain attributes. While invoking this method a callback function will run.

5. onClick()

This method is used to see that the listener that has been registered is triggered by user interaction with the item in the UI.

6. onClick() isEmpty()

This method is used to see that the listener that has been registered.

This method returns true if the string is null.

Diagram.

TW3	
6	<u>2</u>
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/>	
$6 \cdot 0 + 2 \cdot 0 = 8 \cdot 0$	

TW3	
6	<u>2</u>
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/>	
$6 \cdot 0 - 2 \cdot 0 = 4 \cdot 0$	

TW3	
6	<u>2</u>
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/>	
$6 \cdot 0 * 2 \cdot 0 = 12 \cdot 0$	

TW3	
6	<u>2</u>
<input type="button" value="+"/> <input type="button" value="-"/> <input type="button" value="*"/> <input type="button" value="/"/>	
$6 \cdot 0 / 2 \cdot 0 = 3 \cdot 0$	

7. `getText()`

This method returns the text that TextView is displaying

8. `getId()`

This method returns the view's identifier

9. `setText()`

This method sets the text to be displayed.

Source Code

Activity_main.xml

```
<?xml version = "1.0" encoding = "utf-8" ?>
<LinearLayout
    xmlns:android = "http://schemas.android.com/apk/res/android"
    android:orientation = "vertical"
    android:layout_width = "match_parent"
    android:layout_height = "match_parent"
    android:layout_margin = "20dp">
```

<LinearLayout

```
    android:id = "@+id/linearLayout1"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:layout_margin = "20dp">
```

<EditText

```
    android:id = "@+id/editText1"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:layout_weight = "1"
    android:inputType = "numberDecimal"
    android:textSize = "20sp" />
```

</LinearLayout>

<LinearLayout

```
    android:id = "@+id/linearLayout2"
    android:layout_width = "match_parent"
    android:layout_height = "wrap_content"
    android:layout_margin = "20dp">
```

<Button

```
    android:id = "@+id/Add"  
    android:layout_width = "match_parent"  
    android:layout_height = "wrap_content"  
    android:layout_weight = "1"  
    android:text = "+"  
    android:textsize = "30sp" />
```

<Button

```
    android:id = "@+id/Sub"  
    android:layout_width = "match_parent"  
    android:layout_height = "wrap_content"  
    android:layout_weight = "1"  
    android:text = "-"  
    android:textsize = "30sp" />
```

<Button

```
    android:id = "@+id/Mul"  
    android:layout_width = "match_parent"  
    android:layout_height = "wrap_content"  
    android:layout_weight = "1"  
    android:text = "x"  
    android:textsize = "30sp" />
```

<Button

```
    android:id = "@+id/Div"  
    android:layout_width = "match_parent"  
    android:layout_height = "wrap_content"  
    android:layout_weight = "1"  
    android:text = "/"  
    android:textsize = "30sp" />
```

```
<LinearLayout>
```

```
<TextView
```

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

```
    android:layout_width="match_parent"
```

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

```
    android:layout_marginTop="50dp"
```

```
    android:text="Answer is"
```

```
    android:textSize="30sp"
```

```
    android:gravity="center" />
```

```
</LinearLayout>
```

```
MainActivity.java
```

```
package com.example.devang.exno3;
```

```
import android.os.Bundle;
```

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

```
import android.text.TextUtils;
```

```
import android.view.View;
```

```
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity
```

```
    implements OnClickListener
```

```
{
```

```
    EditText Num1;
```

```
    EditText Num2;
```

```
    Button Add, Sub, Mul, Div;
```

```
    TextView Result;
```

@override

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

Num1 = (EditText) findViewById(R.id.editText1);

Num2 = (EditText) findViewById(R.id.editText2);

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

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

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

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

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

Add.setOnClickListener(this);

Sub.setOnClickListener(this);

Mul.setOnClickListener(this);

Div.setOnClickListener(this);

@Override

public void onClick(View v)

float num1 = 0;

float num2 = 0;

float result = 0;

String oper = "";

if (TextUtils.isEmpty(Num1.getText().toString()) ||
TextUtils.isEmpty(Num2.getText().toString()))
return;

num1 = Float.parseFloat(Num1.getText().toString());
num2 = float.parseFloat(Num2.getText().toString());

switch(v.getId())

{

case R.id.Add:

oper = "+";

result = num1 + num2;

break;

case R.id.Sub:

oper = "-";

result = num1 - num2;

break;

case R.id.Mul:

oper = "*";

result = num1 * num2;

break;

case R.id.Div:

oper = "/";

result = num1 / num2;

break;

default

break;

}

Result.setText(num1 + " " + oper + " " + num2 + " = "
+ result);

y

k

Expected Output

- A android application that contains ~~text~~ edit text for two numbers and has to be created. and four buttons for addition, multiplication, subtraction, Division has to be created.
- when user types number and clicks on any of the operator button respective operation should be executed and answer has to be displayed.

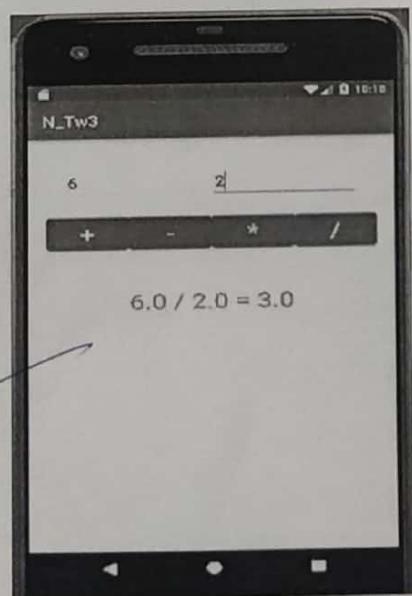
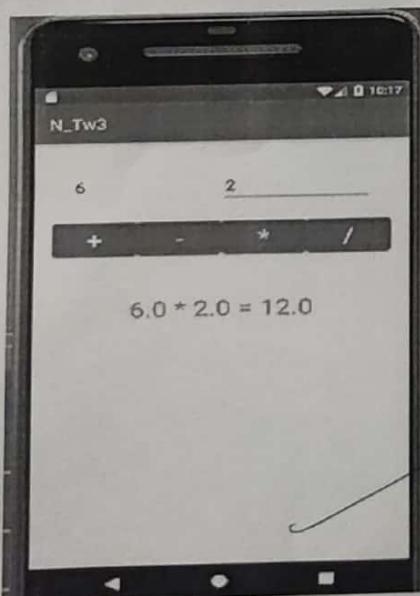
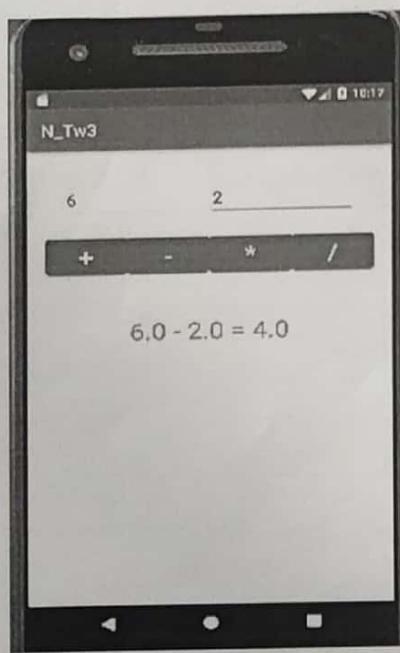
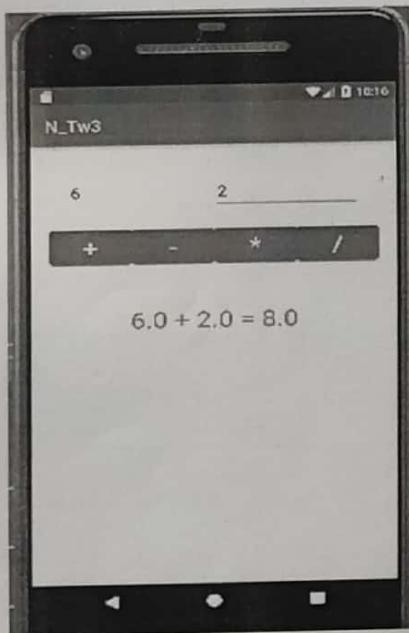
TERMWORK 3

USN: 2GI19CS077

Name: Namrata S Shivakale

Date: 28-11-2022

OUTPUT:



B
28/11/22

Conclusion.

By this framework, we understood the use of linear layout, placing components inside a view, using event listeners in Android Studio to develop a simple calculator Application.

References:

Android Studio 3.5 Development Essentials, Java Edition, 2019 Neil Smyth | Payload Media, Inc.