

Linear, Frame, Constraint calculator app

0. Create a linear layout with 2 buttons (constraint and Frame)

follow along with class demo

for buttons demo

```
android:layout_weight="1"
```

```
android:gravity='right'
```

```
android:layout_gravity="center"
```

add onclick handlers that launch 2 other activities

show on board how this will work with the different layouts, do linear and frame first

1. Create the XML layout using constraint

```
<?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/cost_of_service"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:hint="Cost of Service"
        android:inputType="numberDecimal"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <!-- Will be below cost_of_service field-->
    <TextView
        android:id="@+id/service_question"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="How was the service?"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/cost_of_service" />

    <RadioGroup
        android:id="@+id/tip_options"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checkedButton="@id/option_twenty_percent"
        android:orientation="vertical"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/service_question">
        <RadioButton
            android:id="@+id/option_twenty_percent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Amazing (20%)" />

        <RadioButton
            android:id="@+id/option_eighteen_percent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Good (18%)" />

        <RadioButton
            android:id="@+id/option_fifteen_percent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OK (15%)" />
    </RadioGroup>

    <!--notice layout_width=0dp? makes switch span the parent viewgroup-->
    <Switch
        android:id="@+id/id_roundup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Round up tip?"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@id/tip_options"
        app:layout_constraintTop_toBottomOf="@id/tip_options" />

    <Button
        android:id="@+id/calculate_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:text="Calculate"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/id_roundup" />
    <TextView
        android:id="@+id/tip_result"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tip Amount"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@id/calculate_button" />

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

2. Next gradle

```
//you cannot use a Binding until you add the following lines to the
// Android section of the build.gradle(Module) file (under the
// compileSDK line is fine) and then resync gradle files
//  buildFeatures {
//      viewBinding = true
//  }
```

See:<https://stackoverflow.com/questions/76430646/constant-expression-required-when-trying-to-create-a-switch-case-block>

Starting with Android Gradle Plugin 8.0.0, by default, your resources (e.g. R.id. ...) are no longer declared final (i.e. constant expressions) for optimized build speed, which is a prerequisite to be used in switch statements:

If you want to keep the old behavior, you can add this line in the **gradle.properties** file:

```
android.nonFinalResIds=false
```

If you want to fix it by converting it to if/else statements, Android Studio will help you with

Ctrl + 1 or

Alt + Enter

on the switch keyword.

3. Next the java

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

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

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

```
import android.widget.Toast;
```

```
//import java.lang.Math;
```

```
import com.example.myapplication.databinding.ActivityMainBinding;
```

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

```
// The name of the binding class is generated by converting the  
// name of the XML file to Pascal case and adding the word "Binding"  
// to the end. Similarly, the reference for each view is generated  
// by removing underscores and converting the view name to camel  
// case. For example, in Pascal case activity_main.xml becomes  
// ActivityMainBinding, and you can access @id/text_view as  
// binding.textView.
```

```
private ActivityMainBinding binding;
```

```
@Override
```

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

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

```
    binding = ActivityMainBinding.inflate(getLayoutInflater());
```

```
    setContentView(binding.getRoot());
```

```
//use above instead of the following
```

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

```
//add an anonymous onclick listener (as opposed to doing it in XML)
```

```
binding.calculateButton.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View view) {
```

```
        calculateTip();
```

```
    }
```

```
});
```

```
}
```

```
private void calculateTip() {
```

```
//wrap in a try in case user tries to calc tip with no cost entered
```

```
    try {
```

```
        double cost = Double.parseDouble(binding.costOfService.getText().toString());
```

```
//get selected radio button
```

```
        int id = binding.tipOptions.getCheckedRadioButtonId();
```

```
//get tip amount
```

```
        Double tip = 0.0;
```

```
        switch(id){
```

```
            case R.id.option_twenty_percent:
```

```
                tip = 0.2;
```

```
            break;
```

```

        case R.id.option_eighteen_percent:
            tip=0.18;
            break;
        default:
            tip=0.15;
    }
    Double tots = cost*tip;

    //do we round up?
    if(binding.idRoundup.isChecked())
        tots=Math.ceil(tots);

    binding.tipResult.setText(String.valueOf(tots));
} catch (RuntimeException ex){
    Toast.makeText(MainActivity.this,"Please enter a Cost!",Toast.LENGTH_SHORT).show();
    return;
}
}
}

```

4. Add another activity(*Empty Views Activity*)

```
public class MainActivity3 extends AppCompatActivity {
    boolean bcolor=false;

    RelativeLayout fl;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
    }

    public void doclick(View view) {
        if(fl==null)
            fl = findViewById(R.id.fl);
        bcolor=!bcolor;

        if(bcolor)
            fl.setBackgroundColor(0xFFFF0000);
        else
            fl.setBackgroundResource(R.drawable.ic_launcher_foreground);
    }
}
```

5. setup its 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=".MainActivity3"
  android:background="@drawable/ic_launcher_foreground"
  android:id="@+id/fl"
>
```

```
<TextView
  android:id="@+id/textView"
  android:layout_width="match_parent"
  android:layout_height="120dp"
  android:background="#33262626"
  android:text="TextView"
  android:layout_gravity="bottom"/>
```

```
<Button
  android:id="@+id/button4"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Button"
  android:layout_gravity="bottom|right"
  android:onClick="doclick"/>
```

```
</FrameLayout>
```


6. Call it from main activity

```
<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:text="do frame"
    android:onClick="doclick"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
public void doclick(View view) {
    Intent mi = new Intent(this, MainActivity3.class);
    startActivity(mi);
}
```