FEU-EAST ASIA COLLEGE
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

INFORMATION TECHNOLOGY EDUCATION DEPARTMENT

ITMA133

(MOBILE APPLICATIONS DEVELOPMENT 1)

EXERCISE

3

Simple Calculator

<STUDENT NAME>
<SECTION>
<DATE>

I. OBJECTIVES

At the end of this exercise, students must be able to:

Cognitive

a.) Understand the topics they have learned from lesson 3.

.

Psychomotor:

a.) Design android application with background image.
b.) Process data for computation.
c.) Apply data conversion.
d.) Construct a responsive design for data output.
e.) Apply numeric InputType for EditText.

Affective

a.) Appreciate the concept behind this exercise.

II. BACKGROUND INFORMATION

In order to accomplish this exercise, the student must have a clear understanding of the following topics:

- Data casting
- Java standard methods
- Event handling
- Arithmetic operators

III. LABORATORY PROCEDURE

Overview

This programming exercise demonstrates the use arithmetic operators apply some conditions for validation that will process two input numbers given the operators that was selected.

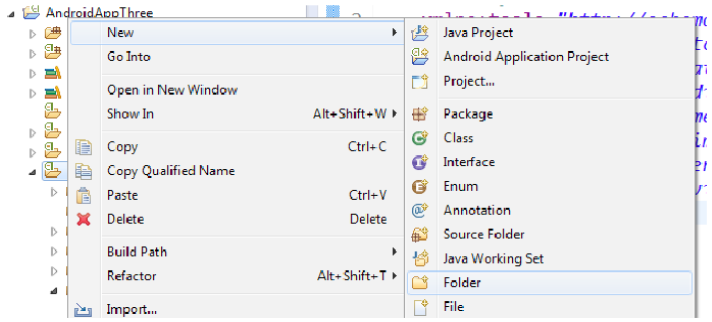
TASK

1. Create a new Android Project.

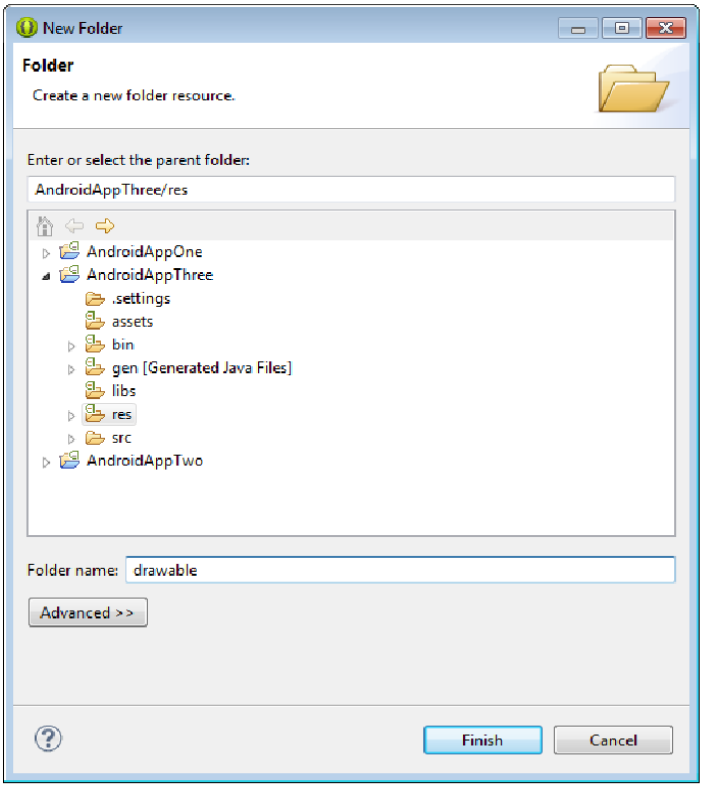
   Project Name: AndroidAppThree
   Activity Name (Main): MainActivity

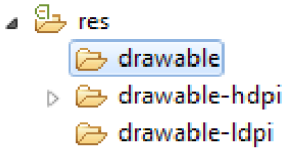
2. Adding background image

   a. Create a new folder named as drawable inside the res folder. (right-click res folder – select new – select folder)

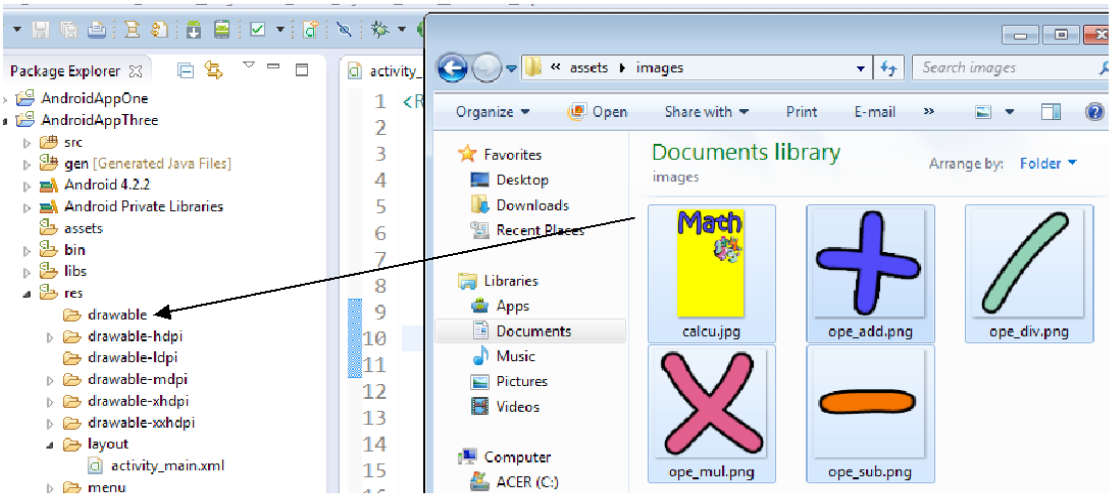


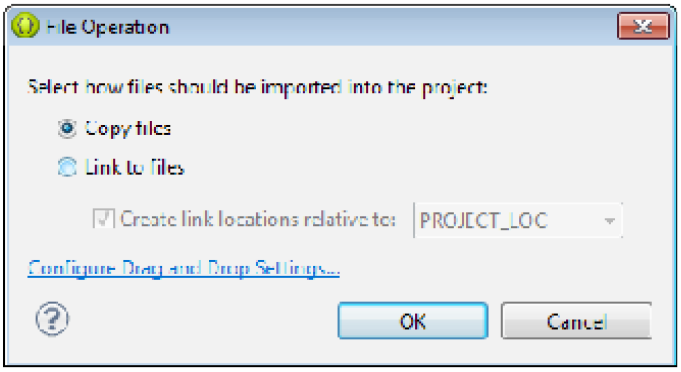
   b. Name the folder as drawable then click finish.

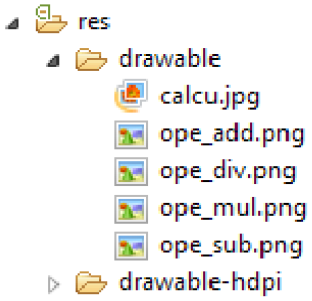




c. Drag and drop the calcu.jpg file from the assets/image folder to res/drawable folder on your project (select copy files).





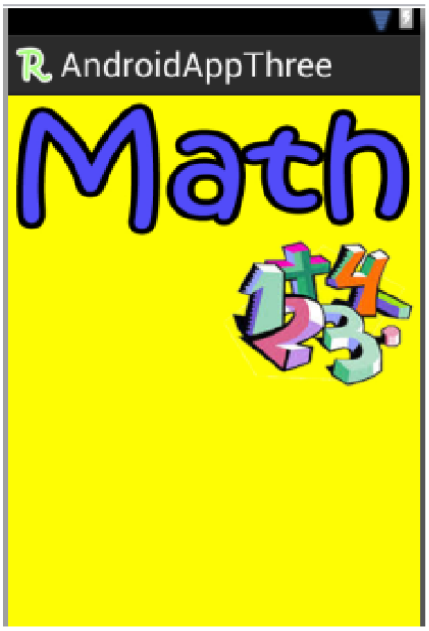


d. Open the activity_main.xml in code view and add the given statement below.

activity_main.xml

```
1  <RelativeLayout xmlns:android="http://schem
2      xmlns:tools="http://schemas.android.com
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:paddingBottom="@dimen/activity_
6      android:paddingLeft="@dimen/activity_ho
7      android:paddingRight="@dimen/activity_h
8      android:paddingTop="@dimen/activity_ver
9      tools:context=".MainActivity"
10     android:background="@drawable/calcu"
11     >
12
13 </RelativeLayout>
```

e. Switch to design view.



3. Constructing design controls
   a. Add xml statement for string value reference as shown below. (res/values/strings.xml)

```
<string name="first_num">First</string>
<string name="second_num">Second</string>
<string name="addition">Addition</string>
<string name="subtraction">Subtraction</string>
<string name="multiplication">Multiplication</string>
<string name="division">Division</string>
<string name="result">0</string>
```

   b. Design the layout (activity_main.xml) as shown below.

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/calcu"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <FrameLayout
        android:id="@+id/frameLayout1"
        android:layout_width="150dp"
        android:layout_height="110dp"
        android:layout_alignParentTop="true"
        android:layout_marginTop="100dp"
        android:background="#0000" >

        <EditText
            android:id="@+id/etNum1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:hint="@string/first_num"
            android:inputType="numberDecimal"
            android:textSize="25sp" >

            <requestFocus />
        </EditText>

        <EditText
            android:id="@+id/etNum2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="60dp"
            android:hint="@string/second_num"
            android:inputType="numberDecimal"
            android:textSize="25sp" />
    </FrameLayout>

    <ImageButton
        android:id="@+id/ibtnAdd"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignLeft="@+id/frameLayout1"
        android:layout_below="@+id/frameLayout1"
        android:layout_marginTop="10dp"
```

```
        android:background="#0000"
        android:contentDescription="@string/addition"
        android:src="@drawable/ope_add"
        android:onClick="doAdd"/>

    <ImageButton
        android:id="@+id/ibtnSub"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignTop="@+id/ibtnAdd"
        android:layout_toRightOf="@+id/ibtnAdd"
        android:background="#0000"
        android:contentDescription="@string/subtraction"
        android:src="@drawable/ope_sub"
        android:onClick="doSub" />

    <ImageButton
        android:id="@+id/ibtnMul"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignTop="@+id/ibtnSub"
        android:layout_toRightOf="@+id/ibtnSub"
        android:background="#0000"
        android:contentDescription="@string/multiplication"
        android:src="@drawable/ope_mul"
        android:onClick="doMul" />

    <ImageButton
        android:id="@+id/ibtnDiv"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_alignLeft="@+id/ibtnMul"
        android:layout_below="@+id/ibtnMul"
        android:background="#0000"
        android:contentDescription="@string/division"
        android:src="@drawable/ope_div"
        android:onClick="doDiv"/>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="340dp"
        android:layout_marginLeft="30dp"
        android:text="@string/result"
        android:textSize="50sp" />

</RelativeLayout>
```

4. Write the java code for the application as shown below.

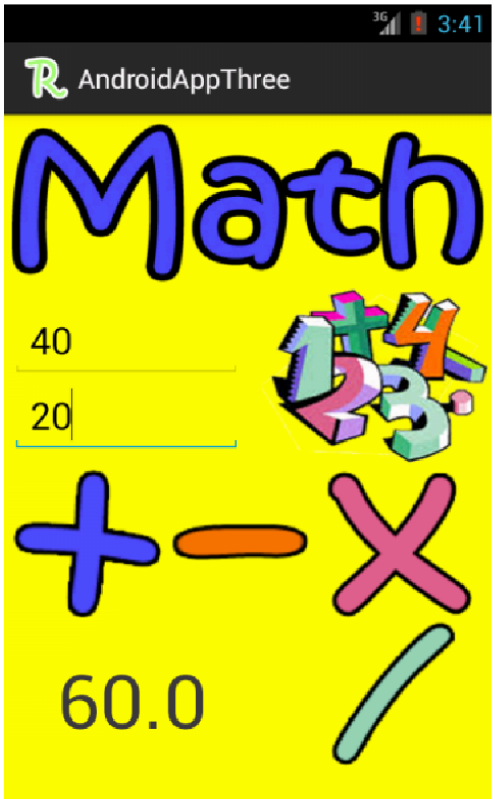
MainActivity.java

```
package com.example.androidappthree;

import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.app.Activity;

public class MainActivity extends Activity {

   TextView tvResult;
   EditText etNum1, etNum2;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      etNum1 = (EditText) findViewById(R.id.etNum1);
      etNum2 = (EditText) findViewById(R.id.etNum2);
      tvResult = (TextView) findViewById(R.id.tvResult);
   }

   public void doAdd(View v){
      float num1, num2, res;
      num1 = getValue(R.id.etNum1);
      num2 = getValue(R.id.etNum2);
      res = num1 + num2;
      tvResult.setText(String.valueOf(res));
   }

   public void doSub(View v){
      float num1, num2, res;
      num1 = getValue(R.id.etNum1);
      num2 = getValue(R.id.etNum2);
      res = num1 - num2;
      tvResult.setText(String.valueOf(res));
   }

   public void doMul(View v){
      float num1, num2, res;
      num1 = getValue(R.id.etNum1);
      num2 = getValue(R.id.etNum2);
      res = num1 + num2;
      tvResult.setText(String.valueOf(res));
   }
```

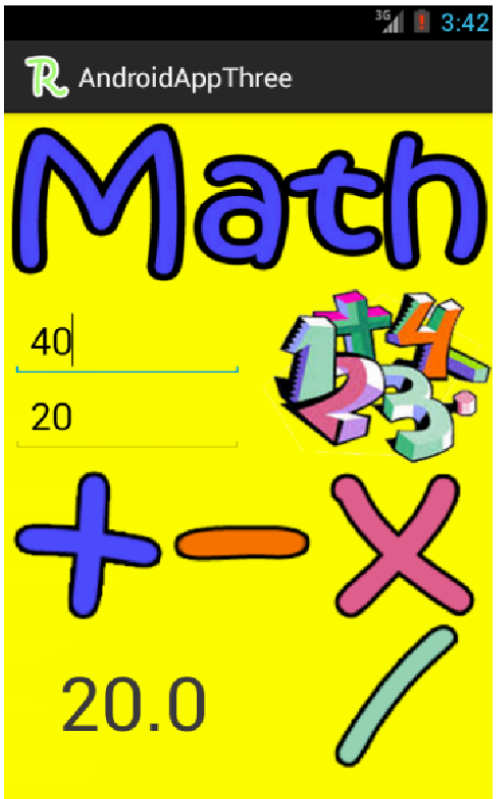
```
    public void doDiv(View v){
        float num1, num2, res;
        num1 = getValue(R.id.etNum1);
        num2 = getValue(R.id.etNum2);
        res = num1 / num2;
        tvResult.setText(String.valueOf(res));
    }

    public float getValue(int id){
        if(id == R.id.etNum1){
            return Float.parseFloat(etNum1.getText().toString());
        } else {
            return Float.parseFloat(etNum2.getText().toString());
        }
    }


}
```

5. Run and test the application

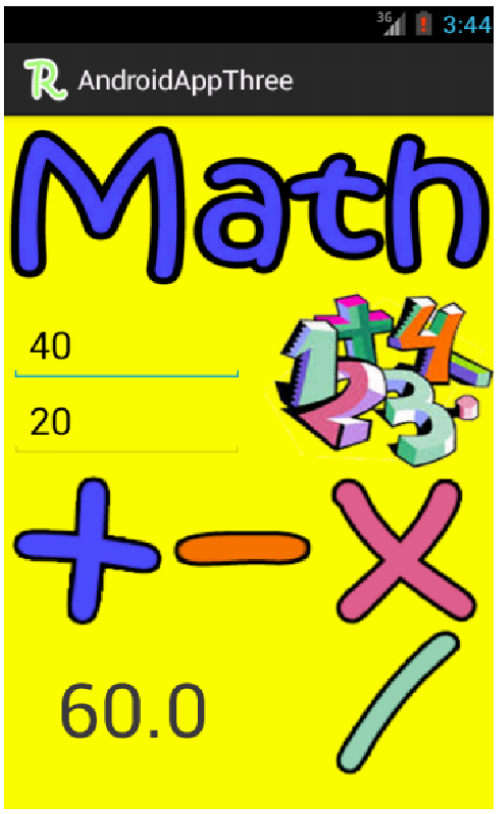
Program Output 1 (Addition):



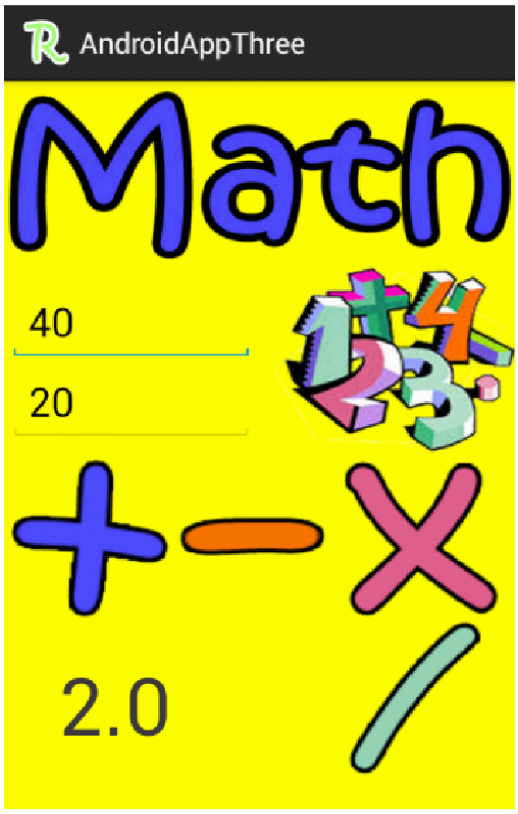
Program Output 2 (Subtraction):



Program Output 3: (Multiplication)



Program Output 4: (Division)



IV. QUESTION AND ANSWER

1. Exception Handling is essential in Java programming. Given that the data types used is integer how do we apply exception handling in our program and what are the expected exception that needs to catch.

2. How do we process all inputted values from the activity especially numbers that needs some arithmetic operations.

3. What are the other inputted types that we can use for edit text. Define their uses.

4. Can we reference on one method all of the operators that we have used in this exercise? How?

V. REFERENCE

http://www.developer.android.com