[(2024A) COMP-5450-AA - Mobile](https://mycourselink.lakeheadu.ca/d2l/home/140923)

Midterm

Aayush Parekh |1215791

<https://github.com/aayushparekhh/Midterm>

QUESTION 1 – CALCULATOR APP

# UI XML Code

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

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="0dp"

android:padding="0dp"

tools:context=".MainActivity">

<TableLayout

android:id="@+id/tableLayout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:stretchColumns="0,1,2,3"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent">

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button\_clear"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onClearClick"

android:text="@string/button\_clear"

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

<Button

android:id="@+id/button\_root"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_root"

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

<Button

android:id="@+id/button\_pow"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_pow"

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

<Button

android:id="@+id/button\_plus"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_plus"

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

</TableRow>

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button\_seven"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_7"

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

<Button

android:id="@+id/button\_eight"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_8"

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

<Button

android:id="@+id/button\_nine"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_9"

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

<Button

android:id="@+id/button\_minus"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_minus"

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

</TableRow>

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button\_four"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_4"

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

<Button

android:id="@+id/button\_five"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_5"

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

<Button

android:id="@+id/button\_six"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_6"

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

<Button

android:id="@+id/button\_multiply"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_mul"

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

</TableRow>

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button\_one"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_1"

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

<Button

android:id="@+id/button\_two"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_2"

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

<Button

android:id="@+id/button\_three"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_3"

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

<Button

android:id="@+id/button\_divide"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_div"

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

</TableRow>

<TableRow

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/button\_dot"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_dot"

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

<Button

android:id="@+id/button\_zero"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@drawable/button\_bg\_stroke"

android:onClick="onNumClick"

android:text="@string/button\_0"

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

<Button

android:id="@+id/button\_equals"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorPrimaryDark"

android:onClick="onEqualsClick"

android:text="@string/button\_equals"

android:textAppearance="@style/TextAppearance.AppCompat.Large"

android:textColor="@android:color/white" />

<Button

android:id="@+id/button\_mod"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="@dimen/box\_size"

android:layout\_height="@dimen/box\_size"

android:background="@color/colorAccent"

android:onClick="onOperatorClick"

android:text="@string/button\_mod"

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

</TableRow>

</TableLayout>

<ImageButton

android:id="@+id/button\_backspace"

style="@style/Widget.AppCompat.Button.Borderless"

android:layout\_width="52dp"

android:layout\_height="40dp"

android:layout\_marginBottom="8dp"

android:layout\_marginEnd="8dp"

android:contentDescription="@string/backspace"

android:onClick="onBackspaceClick"

android:src="@drawable/ic\_backspace\_black\_24dp"

android:textAppearance="@style/TextAppearance.AppCompat.Large"

app:layout\_constraintBottom\_toTopOf="@+id/tableLayout"

app:layout\_constraintEnd\_toEndOf="parent" />

<LinearLayout

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:gravity="end"

android:orientation="horizontal"

app:layout\_constraintBottom\_toTopOf="@+id/button\_backspace"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent">

<TextView

android:id="@+id/input\_value\_1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_marginEnd="8dp"

android:baselineAligned="false"

android:ellipsize="end"

android:maxLength="12"

android:maxLines="1"

android:textAlignment="textEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"

android:textColor="@android:color/black"

app:layout\_constraintBottom\_toTopOf="@+id/button\_backspace"

app:layout\_constraintEnd\_toStartOf="@+id/input\_operation" />

<TextView

android:id="@+id/input\_operation"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_marginEnd="8dp"

android:baselineAligned="false"

android:ellipsize="end"

android:maxLines="1"

android:maxLength="1"

android:paddingLeft="2dp"

android:paddingRight="2dp"

android:textAlignment="center"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"

android:textColor="@color/colorPrimaryDark"

app:layout\_constraintBottom\_toTopOf="@+id/button\_backspace"

app:layout\_constraintEnd\_toStartOf="@+id/input\_value\_2" />

<TextView

android:id="@+id/input\_value\_2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:layout\_marginEnd="24dp"

android:baselineAligned="false"

android:ellipsize="end"

android:gravity="end"

android:maxLength="12"

android:maxLines="1"

android:textAlignment="textEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display1"

android:textColor="@android:color/black"

app:layout\_constraintBottom\_toTopOf="@+id/button\_backspace"

app:layout\_constraintEnd\_toEndOf="parent" />

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_weight="1"

android:gravity="end"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<TextView

android:id="@+id/textView\_result"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginEnd="8dp"

android:layout\_marginTop="8dp"

android:gravity="end"

android:maxLines="1"

android:textAlignment="viewEnd"

android:textAppearance="@style/TextAppearance.AppCompat.Display2"

android:textColor="@android:color/black"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</LinearLayout>

<LinearLayout

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:gravity="end"

android:orientation="vertical"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/linearLayout">

<TextView

android:id="@+id/complete\_operation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginTop="8dp"

android:gravity="end"

android:maxLines="1"

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

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView\_result" />

</LinearLayout>

</android.support.constraint.ConstraintLayout>aa</androidx.constraintlayout.widget.ConstraintLayout>

1. Main Activity Kotlin File

package me.mahakagg.calculator;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

/\*

\* Author: Mahak Aggarwal

\* Last date modified: 26-09-2018

\* (C), All rights reserved, Mahak Aggarwal 2018

\*

\* App Icon -

\* credits to Font Awesome and MDBootstrap for the application logo

\* Link - https://mdbootstrap.com/tools/logo-generator-icons/

\* \*/

public class MainActivity extends AppCompatActivity {

private Calculator mCalculator; // object of Calculator class

private TextView mInputValue1TextView; // for TextView ID - input\_value\_1

private TextView mInputValue2TextView; // for TextView ID - input\_value\_2

private TextView mOperatorTextView; // for TextView ID - input\_operation

private TextView mFinalResultTextView; // for TextView ID - textView\_result

private TextView mCompleteOperation; // for TextView ID - complete\_operation

private double number\_one; // first number

private double number\_two; // second number

private String operation\_string; // current operation

private static int MAX\_CHARACTERS = 10;

private enum operator {

ADD, SUB, MUL, DIV, MOD, ROOT, POW, NULL

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mCalculator = new Calculator();

mInputValue1TextView = findViewById(R.id.input\_value\_1); // first number

mInputValue2TextView = findViewById(R.id.input\_value\_2); // second number

mOperatorTextView = findViewById(R.id.input\_operation); //operation

mFinalResultTextView = findViewById(R.id.textView\_result); // final result

mCompleteOperation = findViewById(R.id.complete\_operation); // string containing the numbers and the operation

operation\_string = operator.NULL.name();

// Implementation of saved instance state

if (savedInstanceState != null) {

mInputValue1TextView.setText(savedInstanceState.getString("First\_number", ""));

mInputValue2TextView.setText(savedInstanceState.getString("Second\_number", ""));

mOperatorTextView.setText(savedInstanceState.getString("Operation", ""));

mFinalResultTextView.setText(savedInstanceState.getString("Final\_result", ""));

mCompleteOperation.setText(savedInstanceState.getString("Complete\_operation", ""));

operation\_string = savedInstanceState.getString("Operation\_string", operator.NULL.name());

number\_one = savedInstanceState.getDouble("Number\_one", 0);

number\_two = savedInstanceState.getDouble("Number\_two", 0);

}

}

// implementation of Saved instance state

@Override

protected void onSaveInstanceState(Bundle outState) {

super.onSaveInstanceState(outState);

outState.putString("First\_number", mInputValue1TextView.getText().toString());

outState.putString("Operation", mOperatorTextView.getText().toString());

outState.putString("Second\_number", mInputValue2TextView.getText().toString());

outState.putString("Final\_result", mFinalResultTextView.getText().toString());

outState.putString("Complete\_operation", mCompleteOperation.getText().toString());

outState.putString("Operation\_string", operation\_string);

outState.putDouble("Number\_one", number\_one);

outState.putDouble("Number\_two", number\_two);

}

// method to check where to append the numbers (first or second number)

private void selectTextViewToAppend(String number) {

if (operation\_string.equals(operator.NULL.name())) {

if (mInputValue1TextView.getText().toString().contains(".")) {

MAX\_CHARACTERS++;

}

if (mInputValue1TextView.getText().length() < MAX\_CHARACTERS) {

mInputValue1TextView.append(number);

MAX\_CHARACTERS = 10;

} else {

MAX\_CHARACTERS = 10;

Toast.makeText(this, "Cannot have more than 10 numbers", Toast.LENGTH\_LONG).show();

}

} else {

if (mInputValue1TextView.getText().toString().contains(".")) {

MAX\_CHARACTERS++;

}

if (mInputValue2TextView.getText().length() < MAX\_CHARACTERS) {

mInputValue2TextView.append(number);

MAX\_CHARACTERS = 10;

} else {

MAX\_CHARACTERS = 10;

Toast.makeText(this, "Cannot have more than 10 numbers", Toast.LENGTH\_LONG).show();

}

}

}

// handle operations for numbers

public void onNumClick(View view) {

if (!mFinalResultTextView.getText().toString().equals("")) {

onClearClick(view);

}

switch (view.getId()) {

case R.id.button\_one:

selectTextViewToAppend("1");

break;

case R.id.button\_two:

selectTextViewToAppend("2");

break;

case R.id.button\_three:

selectTextViewToAppend("3");

break;

case R.id.button\_four:

selectTextViewToAppend("4");

break;

case R.id.button\_five:

selectTextViewToAppend("5");

break;

case R.id.button\_six:

selectTextViewToAppend("6");

break;

case R.id.button\_seven:

selectTextViewToAppend("7");

break;

case R.id.button\_eight:

selectTextViewToAppend("8");

break;

case R.id.button\_nine:

selectTextViewToAppend("9");

break;

case R.id.button\_zero:

selectTextViewToAppend("0");

break;

case R.id.button\_dot:

if (operation\_string.equals(operator.NULL.name())) {

if (mInputValue1TextView.getText().toString().contains(".")) {

Toast.makeText(this, "Cannot have more than one decimal point in a number", Toast.LENGTH\_LONG).show();

} else {

mInputValue1TextView.append(".");

}

} else {

if (mInputValue2TextView.getText().toString().contains(".")) {

Toast.makeText(this, "Cannot have more than one decimal point in a number", Toast.LENGTH\_LONG).show();

} else {

mInputValue2TextView.append(".");

}

}

break;

default:

break;

}

}

// handle operations for operators

public void onOperatorClick(View view) {

if (!mInputValue1TextView.getText().toString().equals("")) {

switch (view.getId()) {

case R.id.button\_plus:

operation\_string = operator.ADD.name();

mOperatorTextView.setText("+");

break;

case R.id.button\_minus:

operation\_string = operator.SUB.name();

mOperatorTextView.setText("-");

break;

case R.id.button\_multiply:

operation\_string = operator.MUL.name();

mOperatorTextView.setText("x");

break;

case R.id.button\_divide:

operation\_string = operator.DIV.name();

mOperatorTextView.setText("/");

break;

case R.id.button\_mod:

operation\_string = operator.MOD.name();

mOperatorTextView.setText("%");

break;

case R.id.button\_pow:

operation\_string = operator.POW.name();

mOperatorTextView.setText("^");

break;

case R.id.button\_root:

operation\_string = operator.ROOT.name();

mOperatorTextView.setText("√");

break;

default:

operation\_string = operator.NULL.name();

break;

}

}

else if (view.getId() == R.id.button\_root) {

mInputValue1TextView.setText("1");

operation\_string = operator.ROOT.name();

mOperatorTextView.setText("√");

}

else {

Toast.makeText(this, "Enter first number before operation", Toast.LENGTH\_LONG).show();

}

}

public void onEqualsClick(View view) {

// handle equals click

if (mInputValue1TextView.getText().toString().equals("") || mOperatorTextView.getText().toString().equals("") || mInputValue2TextView.getText().toString().equals("")) {

Toast.makeText(this, "Enter the numbers and the operation", Toast.LENGTH\_LONG).show();

} else {

number\_one = Double.parseDouble(mInputValue1TextView.getText().toString());

number\_two = Double.parseDouble(mInputValue2TextView.getText().toString());

String operation;

switch (operator.valueOf(operation\_string)) {

case ADD:

mFinalResultTextView.setText(String.valueOf(mCalculator.addition(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_plus) + mInputValue2TextView.getText().toString();

break;

case SUB:

mFinalResultTextView.setText(String.valueOf(mCalculator.subtraction(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_minus) + mInputValue2TextView.getText().toString();

break;

case MUL:

mFinalResultTextView.setText(String.valueOf(mCalculator.multiplication(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_mul) + mInputValue2TextView.getText().toString();

break;

case DIV:

try {

mFinalResultTextView.setText(String.valueOf(mCalculator.division(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_div) + mInputValue2TextView.getText().toString();

} catch (IllegalArgumentException e) {

Toast.makeText(this, getString(R.string.error), Toast.LENGTH\_LONG).show();

operation = "";

clearAll();

}

break;

case MOD:

mFinalResultTextView.setText(String.valueOf(mCalculator.modulus(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_mod) + mInputValue2TextView.getText().toString();

break;

case ROOT:

mFinalResultTextView.setText(String.valueOf(mCalculator.squareRoot(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_root) + mInputValue2TextView.getText().toString();

break;

case POW:

mFinalResultTextView.setText(String.valueOf(mCalculator.power(number\_one, number\_two)));

operation = mInputValue1TextView.getText().toString() + getString(R.string.button\_pow) + mInputValue2TextView.getText().toString();

break;

case NULL:

mFinalResultTextView.setText(getString(R.string.error));

operation = "";

break;

default:

operation = "";

break;

}

mCompleteOperation.setText(operation);

clearAll();

}

}

// handle clear click

public void onClearClick(View view) {

clearAll();

mFinalResultTextView.setText("");

mCompleteOperation.setText("");

}

// clearing most values (needed many times, so created a method to reduce code duplication

public void clearAll() {

mInputValue1TextView.setText("");

mOperatorTextView.setText("");

mInputValue2TextView.setText("");

number\_one = 0;

number\_two = 0;

operation\_string = operator.NULL.name();

}

// handle backspace click (the ImageButton) in the layout

public void onBackspaceClick(View view) {

view.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

clearAll();

return false;

}

});

if (!mInputValue2TextView.getText().toString().equals("")) {

backspaceImplementation(mInputValue2TextView);

} else {

if (!mOperatorTextView.getText().toString().equals("")) {

backspaceImplementation(mOperatorTextView);

} else {

if (!mInputValue1TextView.getText().toString().equals("")) {

backspaceImplementation(mInputValue1TextView);

}

}

}

}

private void backspaceImplementation(TextView view) {

String backspace = view.getText().toString();

backspace = backspace.substring(0, backspace.length() - 1);

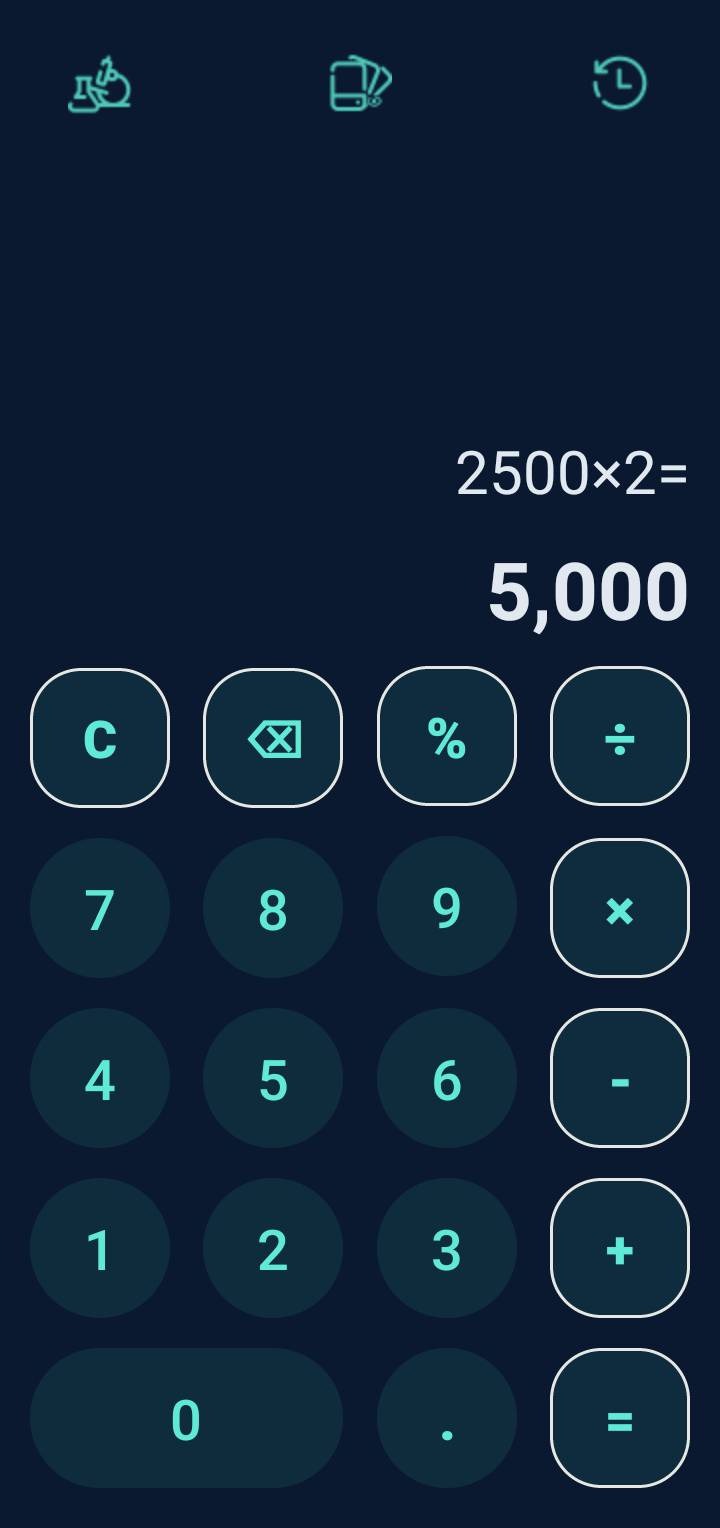
view.setText(backspace);

}

}

# Screenshot





QUESTION 2

# UI XML Code

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

<androidx.coordinatorlayout.widget.CoordinatorLayout 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">

<com.google.android.material.appbar.AppBarLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:theme="@style/AppTheme.AppBarOverlay">

<androidx.appcompat.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary"

app:popupTheme="@style/AppTheme.PopupOverlay"

app:title="@string/app\_name"

/>

</com.google.android.material.appbar.AppBarLayout>

<include layout="@layout/content\_main" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>

1. Main Activity Kotlin File

package com.cvaghela.spinner.searchablespinner

import android.os.Bundle

import android.view.View

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import androidx.lifecycle.MutableLiveData

import com.cvaghela.spinner.searchablespinner.interfaces.OnItemSelectedListener

import com.google.gson.Gson

import com.google.gson.reflect.TypeToken

class MainActivity : AppCompatActivity() {

lateinit var searchableSpinner: SearchableSpinner

lateinit var searchableSpinner1: SearchableSpinner

lateinit var searchableSpinner2: SearchableSpinner

lateinit var searchableSpinner3: SearchableSpinner

lateinit var adapter: SimpleArrayListAdapter

lateinit var adapterTag: SimpleArrayWithTagAdapter

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

searchableSpinner = findViewById(R.id.searchableSpinner)

searchableSpinner1 = findViewById(R.id.searchableSpinner1)

searchableSpinner2 = findViewById(R.id.searchableSpinner2)

searchableSpinner3 = findViewById(R.id.searchableSpinner3)

getListData().observeForever {

if (it.size > 0) {

adapter = SimpleArrayListAdapter(this, it)

adapterTag = SimpleArrayWithTagAdapter(this, it)

searchableSpinner.setAdapter(adapter)

searchableSpinner.setOnItemSelectedListener(object : OnItemSelectedListener {

override fun onItemSelected(view: View?, position: Int, id: Long) {

Toast.makeText(

this@MainActivity,

"Item on position " + position + " : " + adapter.getItem(

position

) + " Selected",

Toast.LENGTH\_SHORT

).show()

}

override fun onNothingSelected() {

Toast.makeText(this@MainActivity, "Nothing Selected", Toast.LENGTH\_SHORT)

.show()

}

})

searchableSpinner1.setAdapter(adapter)

searchableSpinner1.setOnItemSelectedListener(object : OnItemSelectedListener {

override fun onItemSelected(view: View?, position: Int, id: Long) {

Toast.makeText(

this@MainActivity,

"Item on position " + position + " : " + adapter.getItem(

position

) + " Selected",

Toast.LENGTH\_SHORT

).show()

}

override fun onNothingSelected() {

Toast.makeText(this@MainActivity, "Nothing Selected", Toast.LENGTH\_SHORT)

.show()

}

})

searchableSpinner2.setAdapter(adapterTag)

searchableSpinner2.setOnItemSelectedListener(object : OnItemSelectedListener {

override fun onItemSelected(view: View?, position: Int, id: Long) {

Toast.makeText(

this@MainActivity,

"Item on position " + position + " : " + adapterTag.getItem(

position

) + " Selected",

Toast.LENGTH\_SHORT

).show()

}

override fun onNothingSelected() {

Toast.makeText(this@MainActivity, "Nothing Selected", Toast.LENGTH\_SHORT)

.show()

}

})

searchableSpinner3.setAdapter(adapter)

searchableSpinner3.setOnItemSelectedListener(object : OnItemSelectedListener {

override fun onItemSelected(view: View?, position: Int, id: Long) {

Toast.makeText(

this@MainActivity,

"Item on position " + position + " : " + adapter.getItem(

position

) + " Selected",

Toast.LENGTH\_SHORT

).show()

}

override fun onNothingSelected() {

Toast.makeText(this@MainActivity, "Nothing Selected", Toast.LENGTH\_SHORT)

.show()

}

})

}

}

}

fun getListData(): MutableLiveData<ArrayList<String>> {

val encodedString = MutableLiveData<ArrayList<String>>()

val jsonFileString = getJsonDataFromRaw(

this,

R.raw.array\_item

)

val type = object : TypeToken<ArrayList<String>>() {}.type

encodedString.postValue(Gson().fromJson(jsonFileString, type))

return encodedString

}

}

# Screenshot

# ***VIDEO ADDED FOR EASIER DEMONSTRATION***

# Demo SearchableSpinner

QUESTION 3 – Notes app that uses Putrxtra and memory sharing

# UI XML Code

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

<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/layout\_coordinator"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:focusableInTouchMode="true"

tools:context="com.rafapps.simplenotes.NoteActivity">

<ScrollView

android:id="@+id/scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fillViewport="true"

tools:context="com.rafapps.simplenotes.NoteActivity"

tools:showIn="@layout/activity\_note">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<EditText

android:id="@+id/et\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="@string/title"

android:imeOptions="flagNoExtractUi"

android:inputType="textCapWords"

android:maxLength="36"

android:paddingEnd="8dp"

android:paddingStart="8dp"

android:textCursorDrawable="@null"

android:textStyle="bold"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<EditText

android:id="@+id/et\_note"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"

android:autoLink="web|email"

android:ems="10"

android:gravity="top"

android:hint="@string/note"

android:imeOptions="flagNoExtractUi"

android:inputType="textCapSentences|textMultiLine"

android:linksClickable="true"

android:paddingEnd="8dp"

android:paddingStart="8dp"

android:textCursorDrawable="@null"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/title" />

</LinearLayout>

</ScrollView>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

1. Main Activity Kotlin File

package com.rafapps.simplenotes;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.res.ColorStateList;

import android.graphics.Color;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.text.TextUtils;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.EditText;

import androidx.annotation.ColorInt;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.ContextCompat;

import androidx.core.graphics.ColorUtils;

import com.joaomgcd.taskerpluginlibrary.condition.TaskerPluginRunnerCondition;

public class NoteActivity extends AppCompatActivity {

private static final String EXTRA\_NOTE\_TITLE = "EXTRA\_NOTE\_TITLE";

private boolean colourNavbar;

private String title, note;

private EditText noteText, titleText;

private AlertDialog dialog;

private @ColorInt

int colourPrimary, colourFont, colourBackground;

public static Intent getStartIntent(Context context, String title) {

Intent intent = new Intent(context, NoteActivity.class);

intent.putExtra(EXTRA\_NOTE\_TITLE, title);

return intent;

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_note);

titleText = findViewById(R.id.et\_title);

noteText = findViewById(R.id.et\_note);

Intent intent = getIntent();

String action = intent.getAction();

String type = intent.getType();

// If activity started from a share intent

if (Intent.ACTION\_SEND.equals(action) && type != null) {

if ("text/plain".equals(type)) {

String sharedText = intent.getStringExtra(Intent.EXTRA\_TEXT);

noteText.setText(sharedText);

note = sharedText;

title = "";

}

} else { // If activity started from the notes list

title = intent.getStringExtra(EXTRA\_NOTE\_TITLE);

if (title == null || TextUtils.isEmpty(title)) {

title = "";

note = "";

noteText.requestFocus();

if (getSupportActionBar() != null)

getSupportActionBar().setTitle(getString(R.string.new\_note));

} else {

titleText.setText(title);

note = HelperUtils.readFile(NoteActivity.this, title);

noteText.setText(note);

if (getSupportActionBar() != null)

getSupportActionBar().setTitle(title);

}

}

getSettings(PreferenceManager.getDefaultSharedPreferences(NoteActivity.this));

applySettings();

}

@Override

public void onRestart() {

super.onRestart();

note = noteText.getText().toString().trim();

if (getCurrentFocus() != null)

getCurrentFocus().clearFocus();

}

@Override

public void onPause() {

if (!isChangingConfigurations()) {

saveFile();

}

if (dialog != null && dialog.isShowing())

dialog.dismiss();

dialog = null;

super.onPause();

}

@Override

public void onBackPressed() {

super.onBackPressed();

}

@Override

public boolean onSupportNavigateUp() {

onBackPressed();

return true;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_note, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

int id = item.getItemId();

if (id == R.id.btn\_undo) {

noteText.setText(note);

noteText.setSelection(noteText.getText().length());

return (true);

}

if (id == R.id.btn\_share) {

Intent sendIntent = new Intent();

sendIntent.setAction(Intent.ACTION\_SEND);

sendIntent.putExtra(Intent.EXTRA\_TEXT, noteText.getText().toString());

sendIntent.setType("text/plain");

startActivity(Intent.createChooser(sendIntent, getString(R.string.share\_to)));

return (true);

}

if (id == R.id.btn\_delete) {

dialog = new AlertDialog.Builder(NoteActivity.this, R.style.AlertDialogTheme)

.setTitle(getString(R.string.confirm\_delete))

.setMessage(getString(R.string.confirm\_delete\_text))

.setPositiveButton(getString(R.string.yes), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

if (HelperUtils.fileExists(NoteActivity.this, title)) {

deleteFile(title + HelperUtils.TEXT\_FILE\_EXTENSION);

}

title = "";

note = "";

titleText.setText(title);

noteText.setText(note);

finish();

}

})

.setNegativeButton(getString(R.string.no), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

}

})

.setIcon(ContextCompat.getDrawable(getApplicationContext(), R.drawable.ic\_delete\_white\_24dp))

.show();

if (dialog.getWindow() != null) {

dialog.getWindow().getDecorView().setBackgroundColor(colourPrimary);

}

dialog.getButton(DialogInterface.BUTTON\_POSITIVE).setTextColor(Color.WHITE);

dialog.getButton(DialogInterface.BUTTON\_NEGATIVE).setTextColor(Color.WHITE);

return (true);

}

return (super.onOptionsItemSelected(item));

}

private void getSettings(SharedPreferences preferences) {

colourPrimary = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_PRIMARY, ContextCompat.getColor(NoteActivity.this, R.color.colorPrimary));

colourFont = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_FONT, Color.BLACK);

colourBackground = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_BACKGROUND, Color.WHITE);

colourNavbar = preferences.getBoolean(HelperUtils.PREFERENCE\_COLOUR\_NAVBAR, false);

}

private void applySettings() {

HelperUtils.applyColours(NoteActivity.this, colourPrimary, colourNavbar);

// Set text field underline colour

noteText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));

titleText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));

// Set actionbar and background colour

findViewById(R.id.scroll\_view).setBackgroundColor(colourBackground);

if (getSupportActionBar() != null)

getSupportActionBar().setBackgroundDrawable(new ColorDrawable(colourPrimary));

// Set font colours

titleText.setTextColor(colourFont);

noteText.setTextColor(colourFont);

// Set hint colours

titleText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont, 120));

noteText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont, 120));

}

private void saveFile() {

// Get current title and note

String newTitle = titleText.getText().toString().trim().replace("/", " ");

String newNote = noteText.getText().toString().trim();

// Check if title and note are empty

if (TextUtils.isEmpty(newTitle) && TextUtils.isEmpty(newNote)) {

return;

}

// Check if title and note are unchanged

if (newTitle.equals(title) && newNote.equals(note)) {

return;

}

// Get file name to be saved if the title has changed or if it is empty

if (!title.equals(newTitle) || TextUtils.isEmpty(newTitle)) {

newTitle = newFileName(newTitle);

titleText.setText(newTitle);

}

// Save the file with the new file name and content

HelperUtils.writeFile(NoteActivity.this, newTitle, newNote);

// If the title is not empty and the file name has changed then delete the old file

if (!TextUtils.isEmpty(title) && !newTitle.equals(title)) {

deleteFile(title + HelperUtils.TEXT\_FILE\_EXTENSION);

}

// Set the title to be the new saved title for when the home button is pressed

title = newTitle;

// Send Tasker event

TaskerPluginRunnerCondition.Companion.requestQuery(

this,

TaskerEventNoteUpdateActivity.class,

new NoteOutput(newTitle, newNote)

);

}

private String newFileName(String name) {

// If it is empty, give it a default title of "Note"

if (TextUtils.isEmpty(name)) {

name = getString(R.string.note);

}

// If the name already exists, append a number to it

if (HelperUtils.fileExists(NoteActivity.this, name)) {

int i = 1;

while (true) {

if (!HelperUtils.fileExists(NoteActivity.this, name + " (" + i + ")") || title.equals(name + " (" + i + ")")) {

name = (name + " (" + i + ")");

break;

}

i++;

}

}

return name;

}

}

# Screenshot

A screenshot of a cell phone

Description automatically generated

# A screen shot of a phone Description automatically generated

# A screenshot of a phone Description automatically generated

A screen shot of a phone

Description automatically generated