// CALCULATOR IN KOTLIN IN ANDROID STUDIO

**package com.example.layouts**

**import androidx.appcompat.app.AppCompatActivity**

**import android.annotation.SuppressLint**

**import android.os.Bundle**

**import kotlinx.android.synthetic.main.activity\_main.\*  
import net.objecthunter.exp4j.ExpressionBuilder**

**import android.view.View**

**import android.widget.Button**

**import android.widget.EditText**

**class MainActivity : AppCompatActivity()  
{  
  
 override fun onCreate(savedInstanceState: Bundle?)  
 {  
 super.onCreate(savedInstanceState)  
 setContentView(R.*layout*.*activity\_main*) *// buttons are named button(1,2,3,4,5,6,7,8,9,0)***

***and as the users press the keys, that particular***

***number will be passed to editText.***

***button1*.*setOnClickListener* {  
 evaluateExpression("1", clear = true)  
 } // ClickListener is the event used.  
  
 button2.*setOnClickListener* {  
 evaluateExpression("2", clear = true) // number 2 will be pressed and displayed.  
 }  
  
 button3.*setOnClickListener* {  
 evaluateExpression("3", clear = true) // number 3 will be displayed.  
 }**

**button4.*setOnClickListener* {  
 evaluateExpression("4", clear = true)   
 }  
  
 button5.*setOnClickListener* {  
 evaluateExpression("5", clear = true)  
 } // on pressing button named button5  
  
 button6.*setOnClickListener* {  
 evaluateExpression("6", clear = true)   
 }  
  
 button7.*setOnClickListener* {  
 evaluateExpression("7", clear = true)  
 }  
  
 button8.*setOnClickListener* {  
 evaluateExpression("8", clear = true) // on pressing button named button8 on which   
 number 8 is displayed ,8 will be passed to**

**edittext.**

**}  
  
 button9.*setOnClickListener* {  
 evaluateExpression("9", clear = true)  
 }  
  
 button0.*setOnClickListener* {  
 evaluateExpression("0", clear = true) // on pressing button named button0 on which   
 number 0 is displayed ,0 will be passed to**

**edittext.**

**}  
add.*setOnClickListener* {  
 evaluateExpression("+", clear = true)   
 } // add operation will be performed on pressing this   
 button  
 sub.*setOnClickListener* {  
 evaluateExpression("-", clear = true) // subtract operation will be performed on   
 pressing this button  
}  
  
 mul.*setOnClickListener* {  
 evaluateExpression("\*", clear = true) // multiplication operation will be performed**

**on pressing this button.**

**}  
 div.*setOnClickListener* {**

**evaluateExpression("/", clear = true) // divide operation will be performed on   
 pressing this button.**

**}**

**dot.*setOnClickListener* {   
 evaluateExpression(".", clear = true)  
 }  
  
 clear.*setOnClickListener* { // on pressing clear button editText and result will be   
 cleared.**

**edittext.*Text* = ""  
 result.text = ""  
 }   
  
 equalTo.*setOnClickListener* { // on pressing button named equalTo result will**

**be calculated and displayed in edittext result.**

**val text = editText.*text*.toString() // converts to string  
 val expression = ExpressionBuilder(text).build()  
  
 val result = expression.evaluate() // val is to declare variable in KOTLIN   
 val longResult = result.toLong()  
 if (result == longResult.toDouble()) // if-else loop used here**

**{  
 result.text = longResult.toString()  
 } else {  
 result.text = result.toString()   
 }  
 }  
  
 back.*setOnClickListener* { // on pressing button back(DEL)   
 val text = editText.*text*.toString() // converts to string  
 if(text.isNotEmpty()) {  
 editText.*text* = text.drop(1)  
 }  
  
 result.text = ""  
 }  
 }**

**fun evaluateExpression(string: String, clear: Boolean) { //a function is used which is**

**named evaluateExpression**

**if(clear) { // if-else conditional loop used here  
 Result.text = ""  
 Expression.append(string)  
 } else {   
 Expression.append(Result.text)  
 Expression.append(string) // to append expressions  
 Result.text = ""  
 }  
 }  
}**