

A. Main Activity file source code (KOTLIN).

```
private var Numbers = StringBuilder()
private lateinit var calc: TextView
private lateinit var numButtons: Array<Button>
private lateinit var OpButtons: Array<Button>
private var operator: Operators = Operators.NONE
private var Operatorclick: Boolean = false
private var oldnum: Double = 0.0
private var type: Boolean = true

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    Component()
    delete()
    buttonNegPos.setOnClickListener{ NegPosClick() }
    buttonTotal.setOnClickListener{ buttonEqualClick() }
    buttonCLR.setOnClickListener{ (buttonClearClick()) }
}

private fun NegPosClick() {
    if (type) {
        val test = Numbers.toString().toDouble() * -1.0
        Numbers.clear()
        Numbers.append(test)
    } else {
        val test2 = Numbers.toString().toDouble() * 1.0
        Numbers.clear()
        Numbers.append(test2)
    }
    calc.text = Numbers
}

private fun buttonClearClick(){
    Numbers.clear()
    calc.text = "0"
}

private fun delete() {
    buttonDel.setOnClickListener() {
        val length = Numbers.length
        if(length > 1){
            Numbers.deleteCharAt (Numbers.length - 1)
            updateDis()
        }
        else{
            Numbers.clear()
            calc.text = "0"
        }
    }
}
```

```

private fun updateDis() {
    try {
        val textValue = Numbers.toString().toDouble()
        calc.text = textValue.toString()
    }
    catch (e: IllegalArgumentException) {
        Numbers.clear()
        calc.text = "ERROR"
    }
}

private fun Component() {
    calc = findViewById(R.id.calc)
    numButtons =
        arrayOf(button1, button2, button3, button4, button5, button6, button7, button8, button9, button0, buttonDecimal)

    for (i in numButtons) {
        i.setOnClickListener { numButtonsClick(i) }
    }

    OpButtons = arrayOf(buttonAdd, buttonMinus, buttonDivide, buttonMulti)
    for (i in OpButtons) {
        i.setOnClickListener { OperatorButtonsClick(i) }
    }
}

private fun numButtonsClick(btn: Button) {
    if (Operatorclick) {
        oldnum = Numbers.toString().toDouble()
        Numbers.clear()
        Operatorclick = false
    }
    Numbers.append(btn.text)
    updateDis()
}

private fun OperatorButtonsClick(btn: Button) {
    if (btn.text == "+") {
        operator = Operators.ADD
    }
    else if (btn.text == "-") {
        operator = Operators.MINUS
    }
    else if (btn.text == "/") {
        operator = Operators.DIVIDE
    }
    else if (btn.text == "*") {
        operator = Operators.MULTI
    }
    else operator = Operators.NONE
    Operatorclick = true
}

```

```
private fun buttonEqualClick() {
    val newnum = Numbers.toString().toDouble()
    var result: Double
    when(operator){
        Operators.ADD -> result = oldnum + newnum
        Operators.MINUS -> result = oldnum - newnum
        Operators.MULTI -> result = oldnum * newnum
        Operators.DIVIDE -> result = oldnum / newnum
        else -> result = 0.0
    }
    Numbers.clear()
    Numbers.append(result.toString())
    updateDis()
}
```

B. Include a short discussion on how you were able to accomplish the task and.

So, for how I accomplished the task, but first of all I don't think I finished the task 100% because, whenever I inputted a numeric value, it will always show as a double value, it will automatically show the decimal even though I didn't pressed it yet.

In what I did to accomplish the task for the clear button, I made a separate function that clears then return the textview as "0". For delete button I get the length of the String builder and minus it by 1 whenever the delete button was click and I use and if statement that if there is no other numeric value that is greater than 0 in the length it will only show as 0

For the decimal one, I also put it on the number click functions so whenever I press the button of the decimal it will print it in the text view the same way as how the numbers should print.

I have also notice that if you pressed 2 times the decimal it will cause the app to show an error.

- C. Screen Record a short video demo (10-15 seconds) of your working app showing the Clear, Delete, Sign, and Decimal Point functionalities. Create a YouTube channel (if you don't have one yet) and upload your video demo as unlisted. Make sure to include a clickable link of your YouTube video in this report.

<https://www.youtube.com/watch?v=J1AxadK5qJ8>

- D. your self-assessment (what have you learned?).

I have learned how to use and manipulate the difference usage of the different variables and data types of Kotlin, how to use the functions, and to use the if statements or control structure in a project kotlin, the android studio is very similar to the once that I have previously used in my other subjects.