

Name: Christine Ebeo

Student Number: N01455114

Course: ITE-5334 IOS App Dev

Assignment 1

Video URL:

https://drive.google.com/file/d/18NCA0O0_EFTU8epcgCc153Qz5oA1AEoT/view?usp=sharing

Github URL: <https://github.com/tinebeo/ite5334-ios-app-dev>

Calculator:

```
//  
// Calculator.swift  
// calculator  
//  
// Created by Christine Ebeo on 2021-09-21.  
//  
  
import Foundation  
import UIKit  
  
// NOTE: Validations on character input were removed  
//       ViewController controls what user can click on  
class Calculator {  
  
    var statement = [String]()  
    var history = [String]()  
    // standard (default) or advance  
    var mode = CalculatorMode.standard  
  
    // Adds items to statement to use for calculation later  
    func push(s: String) {  
        statement.append(s)  
    }  
  
    // Perform calculation  
    func calc() {  
  
        var result = 0
```

```

var prev: String? = nil

for ch in statement {
    // if character is a number check if calculation is needed
    // else move forward
    if (ch.isInt) {
        // if prev not nil, apply the operand
        if (prev != nil) {
            result = applyOpt(x: result, y: Int(ch)!, opt: prev!)
        } else {
            result = Int(ch)!
        }
    }

    prev = ch
}

// complete the statement, add =result
push(s: "=")
push(s: String(result))

// save history if on advance mode
if (mode == CalculatorMode.advance) {
    saveHistory()
}

}

// Helper to do the calculation
func applyOpt(x: Int, y: Int, opt: String) -> Int {

    var result: Int = 0
    switch opt {
        case "+":
            result = x + y
            break
        case "-":
            result = x - y
            break
        case "*":
            result = x * y
            break
        case "/":
            // output 0 if division by 0

```

```

        if (y == 0) {
            result = 0
        } else {
            result = x / y
        }
        break
    default:
        break
        // nothing to do
    }

    return result
}

// Clean/reset the statement list
func clearStatement() {
    statement = [String]()
}

func getStatement() -> String {
    return statement.joined(separator: "")
}

// Get calculation history
func getHistory() -> String {
    return history.joined(separator: "\n")
}

func saveHistory() {
    // add current statement to history

    history.append(statement.joined(separator: ""))
}

func clearHistory() {
    history = [String]()
}
}

// source : https://stackoverflow.com/questions/38159397/how-to-check-if-a-string-is-an-int-in-swift
extension String {
    var isInt: Bool {
        return Int(self) != nil
    }
}

```

```

    }
}

enum CalculatorMode {
    case standard
    case advance
}

```

ViewController:

```

//
//  ViewController.swift
//  calculator
//
//  Created by Christine Ebeo on 2021-09-21.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var lblOutput: UILabel!
    @IBOutlet weak var historyOutput: UITextView!
    var calculator: Calculator = Calculator()

    // use these for enabling/disabling buttons
    // cant find a way to update buttons using only one variable
    @IBOutlet weak var btn1: UIButton!
    @IBOutlet weak var btn2: UIButton!
    @IBOutlet weak var btn3: UIButton!
    @IBOutlet weak var btn4: UIButton!
    @IBOutlet weak var btn5: UIButton!
    @IBOutlet weak var btn6: UIButton!
    @IBOutlet weak var btn7: UIButton!
    @IBOutlet weak var btn8: UIButton!
    @IBOutlet weak var btn9: UIButton!
    @IBOutlet weak var btn0: UIButton!
    @IBOutlet weak var btnPlus: UIButton!
    @IBOutlet weak var btnMinus: UIButton!
    @IBOutlet weak var btnMultiply: UIButton!
    @IBOutlet weak var btnDivide: UIButton!
    @IBOutlet weak var btnEquals: UIButton!
}

```

```

    lazy var numButtons = [btn1, btn2, btn3, btn4, btn5, btn6, btn7, btn8, btn9,
        btn0, btnPlus, btnMultiply, btnDivide, btnEquals]
    lazy var optButtons = [btnPlus, btnMinus, btnMultiply, btnDivide, btnEquals]

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }

    @IBAction func btnClick(_ sender: UIButton) {

        switch sender.titleLabel!.text! {
            case "C":
                // clear the equation box
                lblOutput.text! = ""
                // activate the numbers and inactivate the operator buttons
                toggleButtons(numIsEnabled: true, optIsEnabled: false)
                // reset the calculator statement list
                calculator.clearStatement()
                break
            case "=":
                // calculate
                calculator.calc()
                // show the resulting statement
                lblOutput.text! = calculator.getStatement()

                // if on advanced, save history
                if (calculator.mode == CalculatorMode.advance) {
                    historyOutput.text = calculator.getHistory()
                }

                // activate the numbers and inactivate the operator buttons
                toggleButtons(numIsEnabled: true, optIsEnabled: false)
                // reset the calculator statement list
                calculator.clearStatement()
                break
            default:
                // numbers or operators were pressed
                // add the current pressed button to the statement list
                calculator.push(s: sender.titleLabel!.text!)
                // activate/inactivate buttons depending on what was pressed
                toggleButtons(numIsEnabled: !btn1.isEnabled, optIsEnabled: !btnPl
us.isEnabled)
                lblOutput.text! = calculator.getStatement()
        }
    }

```

```

}

@IBAction func historySettingBtnClick(_ sender: UIButton) {

    if (calculator.mode == CalculatorMode.standard) {
        sender.setTitle("Standard - No History", for: .normal)

        // when switched back to advance, clear the history output
        calculator.clearHistory()
        historyOutput.text = ""
        calculator.mode = CalculatorMode.advance
    } else {
        sender.setTitle("Advance - With History", for: .normal)
        calculator.mode = CalculatorMode.standard
    }

    // hide/unhide the history area
    historyOutput.isHidden = !historyOutput.isHidden

}

// enable/disables buttons
func toggleButtons(numIsEnabled: Bool, optIsEnabled: Bool) {
    for button in numButtons {
        button!.isEnabled = numIsEnabled
        // updates the button colors
        updateBtnColor(button: button!, color: UIColor.systemOrange)
    }

    for button in optButtons {
        button!.isEnabled = optIsEnabled
        updateBtnColor(button: button!, color: UIColor.systemIndigo)
    }
}

// gray/ungray the buttons
func updateBtnColor(button : UIButton, color: UIColor) {
    if (!button.isEnabled) {
        button.backgroundColor = UIColor.darkGray
    } else {
        button.backgroundColor = color
    }
}
}

```