# --------------------------------------------------------

# Lab 2: Getting Started with Python (In-Lab)

# CIS 103: Introduction to Programming

# Instructor: Md Ali

# Student Name: Trey Wallace

# Date: 8/31/24

# Description:

# This script is a calulator

# --------------------------------------------------------

def add (num1,num2):

    return num1 + num2

def subtract(num1,num2):

    return num1 - num2

def multiply(num1,num2):

    return num1 \* num2

def divide (num1,num2):

    if num2 !=0:

        return num1/num2

    else:

        return "Error ! Division by zero!"

def calculator():

    while True:

        print("1. Add")

        print("2. Subtract")

        print("3. Multiply")

        print("4. Divison")

        print("5. Exit")

        choice = input("Enter Choice ")

        if choice == '5':

            print("exiting, thank you!")

            break

        num1 = float(input("Enter your first number: "))

        num2 = float(input("Enter your second number: "))

        if choice == '1':

            print(add(num1,num2))

        elif choice == '2':

            print(subtract(num1,num2))

        elif choice == '3':

            print(multiply(num1,num2))

        elif choice == '4':

            print(divide(num1,num2))

if \_\_name\_\_ == "\_\_main\_\_":

    calculator()

# --------------------------------------------------------

# Lab 2: Getting Started with Python(Take Home)

# CIS 103: Introduction to Programming

# Instructor: Md Ali

# Student Name: Trey Wallace

# Date: 9/14/24

# Description:

# This script is an enhanced version of the original calculator.py

# Run the code to call the function and follow the prompts to select the desired operation

# Negatives are allowed as well

# --------------------------------------------------------

import math #need to import math from the get go

def add (num1,num2):

    return num1 + num2

def subtract(num1,num2):

    return num1 - num2

def multiply(num1,num2):

    return num1 \* num2

def divide (num1,num2):

    if num2 !=0:

        return num1/num2

    else:

        return "Error ! Division by zero!"#need this to account for zero conundrum

def exponent(num1,num2):

    return num1 \*\* num2

def modulus(num1,num2):

    return num1 % num2

def sqaurert(num1):

    return math.sqrt(num1)

def super\_calculator():

    while True:

        print("Which operation would you like?") #Lets user know they can pick from the below

        print("1. Add")

        print("2. Subtract")

        print("3. Multiply")

        print("4. Divison")

        print("5. Exponent")

        print("6. Modulus")

        print("7. Square Root")

        print("8. Exit")

        choice = input("Enter Choice ")

#These if statements call the functions

#Each choice needs to have the input() within to account for each task and number

#After each operation is complete the code will restart fresh unless the user decides to quit

        if choice == '8':

            print("exiting, thank you!")

            break

        elif choice == '1':

            print("You picked Add")#Reminds the User what they picked just in case

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} + {num2} = {add(num1, num2)}")

        elif choice == '2':

            print("you picked Subtract")

            print("You can also create a negative number, give it a try")

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} - {num2} = {subtract(num1, num2)}")

        elif choice == '3':

            print("You picked Multiply")

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} \* {num2} = {multiply(num1, num2)}")

        elif choice == '4':

            print("You picked Divide")

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} / {num2} = {divide(num1, num2)}")

        elif choice == '5':

            print("You picked Exponent")

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} to the {num2} power is {exponent(num1,num2)}")

        elif choice == '6':

            print("You picked Modulus")

            num1 = float(input("Enter your first number: "))

            num2 = float(input("Enter your second number: "))

            print(f"{num1} % {num2} leaves {modulus(num1,num2)} left")

        elif choice == "7":

            print("You picked Square Root")

            print("Only choose one number this time")

            num1 = float(input("Enter your first number: "))

            print(f"The square root of {num1} is {sqaurert(num1)}")

        else:

            print("Invalid Input Please try again")#this should account for non options

if \_\_name\_\_ == "\_\_main\_\_":

    super\_calculator()

Reflection:

This was a fun but challenging lab. One of the challenges I faced was getting square root to work properly and restart the function. Originally, I had the code set to break after the square root was found. However, since it broke the loop, it did not recall the function so the user could try another option. In tinkering with this I discovered that the way I ordered everything was kind of off. I had the input options outside of the loop and outside of the choices. But also, I treated the square root option as an outlier rather than another elif statement. So, with that discovery I put it at the end of the series of elifs and also put the input options in for each of the elifs. I made square root unique as it already had a function within it thanks to the Math import. From there all that was needed was to add more cosmetic changes to make it easier for the user.

From this lab I learned that order and spacing really does matter in how you build a function. If something resides outside of your if statement it can throw off what you are trying to do or activate/call something too early or even cause your function to recall itself before you or the user are ready to proceed.