

Kimberly Domingo  
Date: 7/9/2023  
Class: IT FDN 110A  
Assignment 02

### Introduction:

Last week, I learned how to collect user input. This week, I will be going through how to use and manipulate these inputs, in this case numbers, to generate an output based on the input. For this program, I will focus on determining the data type of the input and using mathematical operations to manipulate these inputs. Step by step, I will walk through the program I created that (1) ask users to input 2 numbers (2) calculate and print out the sum, difference, product, and the quotient.

### IDEs:

This week, we learned about an IDE (Integrated Development Environment) called Pycharm. This is a software that acts as a text editor, debugger, and a compiler in one. We used IDLE to write our code last week. Both are great editors to write and run Python programs and have many pros and cons. For my program, I will be writing my code in VS Code. This is an IDE that I like. I have realized none are significantly better than others, a lot of it has to do with preferences.

### Writing the Code:

Before I tackle what the assignment is about, I wanted to create an introduction or opening statement that tells users the purpose of the program and what they need to input. This is done by simply using a print function()

```
print(
    """
    Please enter two numbers below and this program will
    calculate them to give you the sum, difference, product, and
    the quotient. Numbers can contain decimals.
    Thank you!!
    """
)
```

**Figure 1: Opening statement using the print() function**

One thing to note is the three quotation marks inside my print() function. The three quotation marks are used when I have a string that spans multiple lines. The way that I have formatted the opening statement in my code will be how it shows up for the users when the code is run.

Next, I will be writing a code that would ask users to input two individual numbers. Like last week, I will use the input() function. These two numbers will be independent of each other and will be changing depending on the user, so I assigned them as two different variables instead of a constant. My two variables are "f\_number" and "s\_number". I use the = to assign these variables to the user input.

```
# Ask users to input two numbers
f_number = input("Enter your first number: ") # First number
s_number = input("Enter your second number: ") # Second number
print(type(f_number)) # check data type before proceeding
```

**Figure 2: User input and code to check the data type of the input**

Before moving forward, I need to check the data type of the user input. There are many different data types, so I need to make sure that the user input is a data type that I can use mathematical operations on. To do this, I use the `type()` function. Inside the parentheses, I put in the variable that I want to check the data type on. I nest this inside the `print()` function, so it will give me the print out the data type.

The data type of the user input value is a string. I will not be able to use mathematical operations on strings, so I need to convert these strings to either a float or integer. Either one would work to convert the strings to numbers, but one is more ideal. To do a simple comparison, integers can be looked at as whole numbers, while floats include both whole numbers and decimals.

Using an integer could cause an Error in the program. For example, if a user wanted to input “3.6” for their first number, it would show an error. Because of this, I will convert the strings into floats to avoid this error in the program.

```
Enter your first number: 3.6
Traceback (most recent call last):
  File "/Users/kim/Documents/_Python Class/Assignment02/BasicMath.py", line 18, in <module>
    f_number = int(input("Enter your first number: "))
ValueError: invalid literal for int() with base 10: '3.6'
```

**Figure 3: Error saying “3.6” is invalid for int()**

Now that users can input decimals if they want to as our variables, they can be used in mathematical equations. The four operations are addition, subtraction, multiplication, and division. I will assign each mathematical operation to a variable. To make it easy for myself, I will use whatever operation I will be performing as the variable name.

```
# Operations to calculate the two numbers
# Operations is the same in python as if you are calculating it with a pen and paper
sum = (f_number + s_number)
difference = (f_number - s_number)
product = (f_number * s_number)
quotient = (f_number / s_number)
```

**Figure 4: Mathematical operations I performed on the inputs**

Even though I have calculated what I needed to, the users cannot see the results since they are not printed out. I will use the `print()` function to print out the results. The parentheses inside the

print function will include a combination of strings and variables that is assigned to the inputted numbers and the result of the mathematical operations.

```
# print out the answers for the user
print("The sum of", f_number, "and ", s_number, "is ", sum)
print("The difference between", f_number, "and", s_number, "is", difference)
print("The product of", f_number, "and", s_number, "is", product)
print("The quotient of", f_number, "and", s_number, "is", quotient)
```

**Figure 5: Commas are used to separate the strings and variables in the print function**

There are many different ways of combining strings and variables in a print function. I used commas as a way to separate these, but a "+" sign is also a common way to do so.

```
                Please enter two numbers below and this program will
                calculate them to give you the sum, difference, product, and
                the quotient. Numbers can contain decimals.
                Thank you!!

Enter your first number: 60.5
Enter your first number: 2.8

The sum of 60.5 and 2.8 is 63.3

The difference between 60.5 and 2.8 is 57.7

The product of 60.5 and 2.8 is 169.39999999999998

The quotient of 60.5 and 2.8 is 21.607142857142858
```

**Figure 6: Output of my program**

Now that my code is done, I run it and this is my end result. I was able to get an user input that may include a decimal and print out the sum, difference, product, and quotient of the two numbers.

After looking at my code, it looks very messy and could be cleaned up a little bit. One way to clean it up is to consolidate the lines of code.

Assigning a variable to the user input and changing the data type to a float is one part that could be more organized. I am working with the same variables for each function.

```
# Ask users to input two numbers
f_number = input("Enter your first number: ") # First number
s_number = input("Enter your second number: ") # Second number
print(type(f_number)) # check data type before proceeding

# Convert data type from a str to a float
f_number = float(f_number) # using the float function to convert the data type of f_number to a float
s_number = float(s_number)
```

**Figure 7: Old, unorganized code**

As you can see in Figure 7: I am manipulating the same variable when changing it to a different data type. I can consolidate this to two lines of code instead of four lines of code. To do this, I

will assign the two variables to a float function. The input function will then be nested inside the float function.

```
5
6 # Instead of writing two addition lines of code to convert the string to a float, you can nest the input
7 # function in the the float function
8 f_number = float(input("Enter your first number: "))
9 s_number = float(input("Enter your first number: "))
```

**Figure 8: A more organized line of code to change the data type of the input to a float**

I can also do the same thing for the line of codes that calculate and print out the results. I will be combining the operations lines of code and the print() function code into one.

```
# Within the print function, I can calculate the two numbers by putting in the operations inside the print function
print("\nThe sum of", f_number, "and ", s_number, "is ", (f_number + s_number))
print("\nThe difference between", f_number, "and", s_number, "is", (f_number - s_number))
print("\nThe product of", f_number, "and", s_number, "is", (f_number * s_number))
print("\nThe quotient of", f_number, "and", s_number, "is", (f_number / s_number))
```

**Figure 9: Inserting the mathematical operations inside the print() function**

Instead of assigning separate variables to perform the computation, I will put the mathematical operations inside the print function. This will still perform the operations as long as our variables are correct.

```
# Clean up the script
# Most of the line of code can be consolidated into one or two line of code

print(
    """
    Please enter two numbers below and this program will
    calculate them to give you the sum, difference, product, and
    the quotient. Numbers can contain decimals.
    Thank you!!
    """
)

# Instead of writing two addition lines of code to convert the string to a float, you can nest the input
# function in the the float function
f_number = float(input("Enter your first number: "))
s_number = float(input("Enter your first number: "))

# Within the print function, I can calculate the two numbers by putting in the operations inside the print function
print("\nThe sum of", f_number, "and ", s_number, "is ", (f_number + s_number))
print("\nThe difference between", f_number, "and", s_number, "is", (f_number - s_number))
print("\nThe product of", f_number, "and", s_number, "is", (f_number * s_number))
print("\nThe quotient of", f_number, "and", s_number, "is", (f_number / s_number))
```

**Figure 10: My complete code**

As you can see in Figure 10, my program contains a cleaner code that still does exactly the same thing as my previous, less organized code.

### Summary:

This program allowed users to input two numbers, including decimals which are assigned to two individual variables. These variables were converted to floats. The sum, difference, product, and the quotient of the two numbers were then calculated and the answers were presented back to the user.

