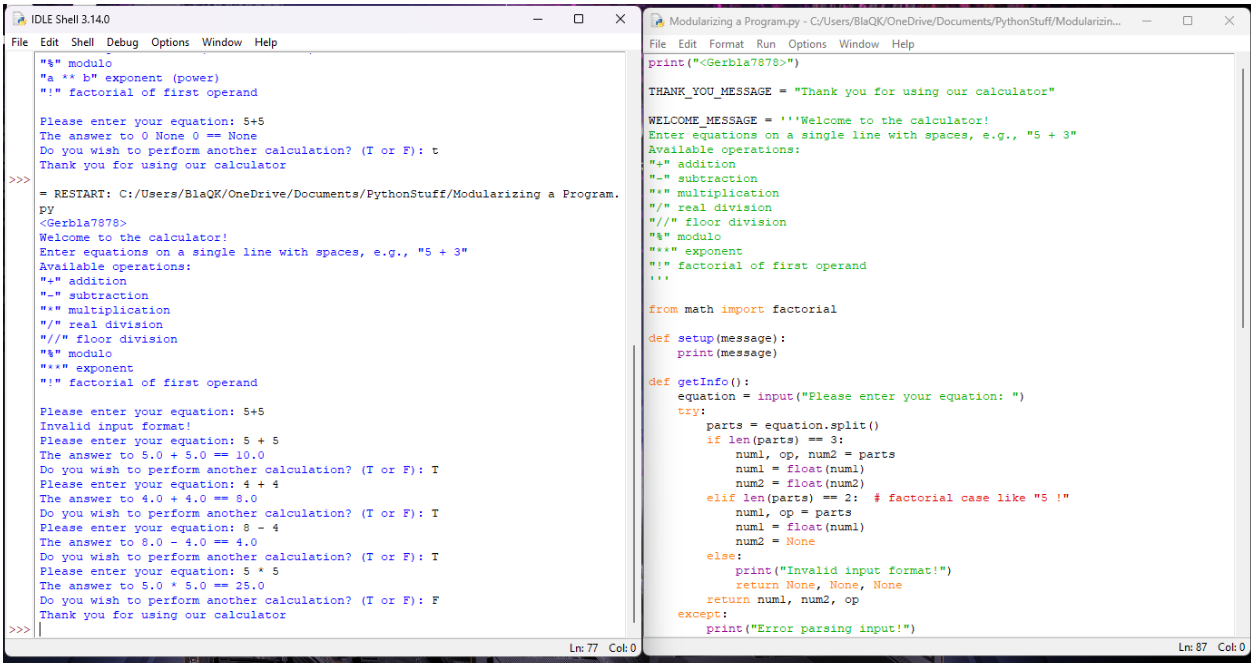
Gerald Blackwell

3.4 Guided Practice Modularizing a Program


IDLE Shell 3.14.0
File Edit Shell Debug Options Window Help
Ln: 77 Col: 0
Modularizing a Program.py - C:/Users/BlaQK/OneDrive/Documents/PythonStuff/Modularizin...
File Edit Format Run Options Window Help
Ln: 87 Col: 0

```
"%" modulo
"a ** b" exponent (power)
"!" factorial of first operand

Please enter your equation: 5+5
The answer to 0 None 0 == None
Do you wish to perform another calculation? (T or F): t
Thank you for using our calculator
>>>
= RESTART: C:/Users/BlaQK/OneDrive/Documents/PythonStuff/Modularizing a Program.
py
<Gerbla7878>
Welcome to the calculator!
Enter equations on a single line with spaces, e.g., "5 + 3"
Available operations:
"+" addition
"-" subtraction
"*" multiplication
"/" real division
"//" floor division
"%" modulo
"**" exponent
"!" factorial of first operand

Please enter your equation: 5+5
Invalid input format!
Please enter your equation: 5 + 5
The answer to 5.0 + 5.0 == 10.0
Do you wish to perform another calculation? (T or F): T
Please enter your equation: 4 + 4
The answer to 4.0 + 4.0 == 8.0
Do you wish to perform another calculation? (T or F): T
Please enter your equation: 8 - 4
The answer to 8.0 - 4.0 == 4.0
Do you wish to perform another calculation? (T or F): T
Please enter your equation: 5 * 5
The answer to 5.0 * 5.0 == 25.0
Do you wish to perform another calculation? (T or F): F
Thank you for using our calculator
>>> |
```

```python
print("<Gerbla7878>")

THANK_YOU_MESSAGE = "Thank you for using our calculator"

WELCOME_MESSAGE = '''Welcome to the calculator!
Enter equations on a single line with spaces, e.g., "5 + 3"
Available operations:
"+" addition
"-" subtraction
"*" multiplication
"/" real division
"//" floor division
"%" modulo
"**" exponent
"!" factorial of first operand
'''

from math import factorial

def setup(message):
    print(message)

def getInfo():
    equation = input("Please enter your equation: ")
    try:
        parts = equation.split()
        if len(parts) == 3:
            num1, op, num2 = parts
            num1 = float(num1)
            num2 = float(num2)
        elif len(parts) == 2:  # factorial case like "5 !"
            num1, op = parts
            num1 = float(num1)
            num2 = None
        else:
            print("Invalid input format!")
            return None, None, None
        return num1, num2, op
    except:
        print("Error parsing input!")
```

1. Explain fully the difference between the equal sign (=) and the double equal sign (==) in Python. **The single equal sign = is used to assign a value to a variable, such as `x = 5`. The double equal sign == is used to compare two values to see if they are equal, returning `True` or `False`, such as `x == 5`. In short, = stores a value, while == checks for equality.**
2. When you use the keyword return in a function, what is it that is returned, and where is it returned? **When you use the keyword `return` in a function, the value or object that follows `return` is sent back to the place where the function was called. This allows the calling code to use that value, store it in a variable, or perform further operations with it.**
3. This requires a little research. In the example above, in the getInfo() function, we use the method number.count(" ") = =2. Explain this syntax. **`number.count(" ") == 2` checks whether the input string `number` contains exactly two spaces. The `count(" ")` method counts how many spaces are in the string, and the `== 2` part tests if that count is equal to 2. This helps the program determine whether the input has three parts (like `5 + 3`) or fewer.**

   https://docs.python.org/3/library/stdtypes.html#str.count

https://docs.python.org/3/reference/expressions.html#comparisons

4. This requires a little research. In the example above, in the getInfo() function, we use the syntax: number1,operation,number2 = number.split(" "). Explain what occurs on this line.
   The line `number1, operation, number2 = number.split(" ")` splits the input string `number` at each space into a list of parts. Then it assigns the first part to `number1`, the second part to `operation`, and the third part to `number2`. This lets the program separate the two numbers and the operator from the user's input so it can perform calculations.

   https://docs.python.org/3/library/stdtypes.html#str.split