

Assignment1_Skeleton(2)

January 14, 2024

ASSIGNMENT 1: CALCULATOR

In this assignment, your challenge is to create a four function calculator using a calculator class containing 4 operator functions (add, subtract, multiply, divide). It should have a constructor, member variables, and member functions.

```
[7]: # To get started, create a Calculator class. It should have the member
      ↪ variables of color, brand and owner which are created in the constructor.

# Let's first define the class using the class keyword, i.e. class ClassName:

#####
# Your Code Starts Here
#####

#define class

class Calculator:

#####
# Your Code Ends Here
#####

# A constructor can be created using def __init__(self, param1, param2, etc.):
      ↪ where param1 and param2 are the member variables.
# The next few lines should set the member variables to the values passed into
      ↪ the constructor, e.g. self.param1 = param1, self.param2 = param2, etc.
# So, our example with two member variables param1 and param2 would be:
#     def __init__(self, param1, param2):
#         self.param1 = param1
#         self.param2 = param2
# Remember that we are still writing the various components of the class. Don't
      ↪ forget to indent!

#####
# Your Code Starts Here
#####
```

```

#define constructor
def __init__(self, color, brand, owner):
    self.color = color
    self.brand = brand
    self.owner = owner

#####
# Your Code Ends Here
#####

# The class should then contain the 4 operator functions which take in
↳numerical values and return the result.
# We've done the add function already for you below.

def add(self, val1, val2):
    return val1 + val2

# Now, create functions for the other 3 variables. Remember to properly indent!

#####
# Your Code Starts Here
#####

#subtract
def subtract(self, val1, val2):
    return val1 - val2

#multiply
def multiply(self, val1, val2):
    return val1 * val2

#divide
def divide(self, val1, val2):
    if val2 == 0:
        return "Error: Division by zero"
    return val1 / val2

#####
# Your Code Ends Here
#####

# Lastly, the class should have a function called do_calculation with only a
↳self parameter.
# This do_calculation function prompts the user to input values and an
↳operator, and prints out the corresponding output to the console.

```

```
#####
# Your Code Starts Here
#####

#do_calculation function
def do_calculation(self):
    #input one
    v1 = int(input("Enter a number: "))
    #input two
    v2 = int(input("Enter a number: "))

    #operator input
    op = input("Enter operator (+, -, x, /): ").lower()

    #if-else statments to run calculations
    #add
    if (op == "+"):
        print(self.add(v1, v2))
    #subtract
    elif (op == "-"):
        print(self.subtract(v1, v2))

    #multiply
    elif (op == "x"):
        print(self.multiply(v1, v2))

    #divide
    elif (op == "/"):
        print(self.divide(v1, v2))

    #invalid input case
    else:
        print("Invalid operation")

#####
# Your Code Ends Here
#####
```

Now let us create an instance of that class we just wrote. This is somewhat similar to creating a new car at an auto plant; think of it as the creation of an object of the car class. An object is *instantiated* as such:

```
object = Class(self, param1, param2, etc.)
```

```
[8]: #####
# Your Code Starts Here
#####
```

```

    #create an instance of the Calculator class
my_calc = Calculator("Black", "Casio", "Jason")

#####
# Your Code Ends Here
#####

```

Now, let's run the `do_calculation` function to run the calculator!

```

[9]: #replace my_calc with whatever name you used to instantiate the object in the
      ↪previous code block.
my_calc.do_calculation()

```

18

Congratulations, you have now used objects and classes to create a fully functioning basic calculator in Python!

Write your report below:

My calculator code declares a calculator class with a constructor that instantiates the calculators color, brand, and owner. It then declares 4 of the basic math functions addition, subtraction, multiplication, and division. Finally, it declares the `do_calculation()` function that prompts the user for two integers and an operation with the possibilities: +, -, x, /. If the user gives an incorrect operation, the function prints an error.

Challenges Incurred: I initially used `if` and `else` states instead of `if`, `elif`, and `else` statements, so even if a user inputted a valid operation, it would produce the correct answer and an error message.