1. KISS - Used a stack and then a for loop to iterate over it, add the numbers assign the addition to a variable, and then return it. Essentially, it is overcomplicating the code and doing unnecessary things.

```
def adding_numbers(num1, num2):
       #using a stack because it is OPTIMALLLLLL
       stack = []
       addition = 0
       stack.append(num1)
       stack.append(num2)

       for i in stack:
            addition += i
       return addition
```

2. Smelly Comments - described the actual logic of the function but used terrible variables names making it hard for others to understand and read.

```
def d(a, b):
       #initialize c to keep return total
       c=0
       #set return value to quotient of two inputs
       c = a / b
       #return return value
       return c
```

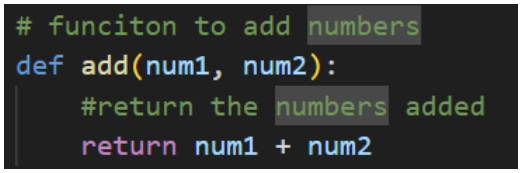
3. YAGNI- Implementing a method that does not serve a function asides from only adding one. This could easily be done with the addition method.

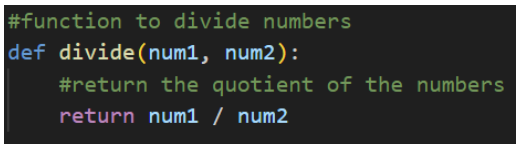
```
def addOne(num1):
       #Anthony definitely made this stupid function. Does he even know how to center
a div?
       return num1 + 2 - 1
```

4. SoC - violates separation of concerns by calling other functions whose purpose is to do one thing(the multiply function calling the adding function). This can lead to confusion and is unnecessary.

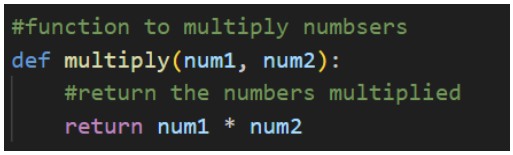
```
#function to multiply numbers
    #violates separation of concerns principle
    def multiply(num1, num2):
        total = 0
        for i in range(num2):
            total = Calculator.adding_numbers(total, num1)
        return total
```

Solutions:

1. 
```
# funciton to add numbers
def add(num1, num2):
    #return the numbers added
    return num1 + num2
```

2. 
```
#function to divide numbers
def divide(num1, num2):
    #return the quotient of the numbers
    return num1 / num2
```

3. Remove the function or don't begin writing it until it is necessary.

4. 
```
#function to multiply numbsers
def multiply(num1, num2):
    #return the numbers multiplied
    return num1 * num2
```