

good calculator:

Single Responsibility Principle:

there are 3 separate functions for calculations:

add(): adds 2 numbers

subtract(): subtracts 2 numbers

division(): performs division, and has a check for division by 0

multiplication: multiplication of 2 numbers

These functions are very simple, reusable, and follow the Single Responsibility Principle. Each function does a specific task and does it well!

KISS:

the code is kept very simple so it can be easy to read. the functions do what is necessary and there is no complex logic for performing calculations.

I decided to use a dictionary to store the operators, so many if else statements are not needed.

DRY code:

I utilized the dictionary operation so I do not have to have 5 if-else statements. Avoiding large code blocks is ideal!

bad calculator:

there are many issues with this code:

there are no functions and all operations are placed in a while loop. this does not follow modular design and separation of concerns

the variable names are bad and not specific enough. a, b, and o are too vague

there is a lack of input validation, and the program will crash if a non-numeric input is given

division by 0 is done incorrectly

no comments or documentation

redundant code, there are many repeated statements and overall redundant code.