Using theses principles: Seperation of concern, document your code, and clean code

Hello, for this activity, I designed an in-terminal calculator, which allows users to perform calculations including addition/subtraction and multiplication/division.

For my properly written code (calculator.c) I focused on clear and concise code with good comments and well separated functions which make the code easily readable, and if an error were to occur it can be quickly located and corrected. I surpassed having three functions, and instead had four, separating addition, subtraction, multiplication, and division into their own functions that can be called by the main function depending on what the user selects in terminal. I then used a switch function which allows for these individual functions to be called and employed. I utilized separation of concerns by separating these smaller functions from the main function, I employed propper documentation of code through comments above each function, and lastly I made sure to remove any unnecessary snippets of code to keep everything concise. For the horrible code (calc.c) I scrapped the documentation entirely, bundled the functions into one main function, and I made sure the code was not clean by intentionally swapping the switch statement to if statements and adding essentially pointless/sloppy separation.