

### How Good-Calculator.py implements the principles:

- **KISS (Keep it simple, stupid):**
  - Each function has a clear operation, with no convoluted logic.
  - For example, `def add(a, b)` just does `a+b`. The main function just accesses user input and if the user enters 1, it adds. Same for all the other functions and operations.
- **Clean Code:**
  - Each function does its own assigned purpose, and the entire code is easy to read, and, therefore, easy to debug.
  - It's just a few functions, accessing user input, and the only error handling is just if the user enters a number that's not in the list, and also allows the user to exit the program. If there are any issues, they'll be easy to spot.
- **Document Your Code:**
  - I added clear explanations of what each function does, and also what the main function does.
  - There are no unnecessary comments, and not every line is commented out – the only comments that are there are for functionality purposes (what does it do/why does it do it) rather than (oh, `x + 1` adds one to `x`).

### How Bad-Calculator violates the principles:

- **DRY (Don't Repeat Yourself)**
  - The program violates the DRY principle by repeating large chunks of code.
  - For example, in the main `run_calculator` function, the `'if op_choice == '+'` block is practically the same as the `'elif op_choice == '-'` block. This makes things tricky as if a change was needed for the input prompt, it would be updated in both places.
- **YAGNI (You Ain't Gonna Need It)**
  - The code includes features that are absolutely unnecessary for a simple calculator that only adds and subtracts.
  - One of the prime features that adds zero value onto the program is the `'UserProfile'` class, which was never used by the program's core logic. They increase the file size and add complexity for no reason, making the code harder to understand.
- **Clean Code**
  - The code is harder to read, understand and debug compared to typical code.
  - There are vague and non-descriptive names for functions such as `"do_calc"` and variables like (`"p"`, `"x1"`, `"x2"`, and `"val1"`), forcing anyone who is reading the code to guess what the variables are for.