Project 1 Question 1 Version 1 documentation

The problem of the code before refactor

Before the refactor, the main problem with the code was that it lacked structure and had a lot of repetitive code. The logic for input, calculations, and printing was all placed within the main function, making it difficult to understand and maintain. One of the issue with the code before the re-factor is: -

Lack of function abstraction: The code directly performed all the input, calculation, and printing tasks within the main function. This made the code long, hard to read, and difficult to follow. It also violated the principle of separating concerns and reusing code.

Readability: By dividing the code into functions with meaningful names, the code becomes more readable and self-explanatory. The main function provides a high-level overview of the program's flow, while the input and calculation functions encapsulate the specific tasks they perform. This makes it easier for other developers (including yourself) to understand the code and its functionality.

The advantage of refactoring the code by using functions

- ✓ Improved code organization: The refactored code separates the input, calculation, and printing logic into different functions. This improves the overall organization of the code and makes it easier to understand and maintain. Each function has a specific purpose, making it easier to locate and modify specific parts of the code if needed.
- ✓ Reusability: By separating the input, calculation, and printing logic into functions, you make them reusable. If you need to perform similar calculations or print similar outputs in another part of your program, you can simply call these functions again without duplicating code. This promotes code reuse and reduces redundancy.
- ✓ Maintainability: The refactored code is easier to maintain and modify. If you need to make changes to the input process, calculation logic, or printing format, you can do so within their respective functions without affecting other parts of the code. This modular structure simplifies maintenance and reduces the likelihood of introducing bugs while making changes.