

Assignment02-Q4:

The self-documentation principle focuses on having clear naming, comments, structure and readability. All my solutions to the 3 questions embody this principle.

- I use clear, descriptive and meaningful variable and method names like `decimalToOctal`, `num_triangle`, and `deci_num` to describe the function/purpose of the variables and methods.
- I use detailed and concise comments at the top of each program describing what it does. Additionally, I used comments to explain more in detail what each section of the code does.
- I used a simple and neat structure for all my solutions such that it is visually pleasing and easy to read.

By using the self-documentation principle in my code, I was able to make my code more readable, structured and understandable to others who mark/read over my code.

I also embodied the encapsulation principle in my code:

- I made sure to use **private** methods in my code. Most of the major calculations and comparisons were done inside these private methods and only the user input and output code was defined in the main (public) method.
- I broke down each solution into methods - this not only allowed me to keep different processes separate but also allowed the code to be more readable, maintainable, and modular.
- This allowed me to control the access of the methods.

Assignment02-Q5:

Encapsulation:

- Allowed me to protect internal processes like calculations and handling sensitive data.
- Allowed me to control the access (public/private) of each method.
- In case I wish to make changes to my code, I can easily change data/certain processes without affecting code in other methods.
- Ex: If I want to change the decimal to octal code to decimal to binary code later, encapsulation allows me to change only the `decimalToOctal` method and not the main method. Additionally, I can still retain the old method and create a new method for `decimalToBinary`
- Drawback of not using encapsulation: The internal processes would be exposed and vulnerable to unwanted changes. This also makes changing/editing the code risky and tiring.

Self-documenting:

- Improves code readability. Code is more understandable to others. This is useful for collaboration.
- The code is more manageable. Programmers can go back to the code after a long and still understand the functions of the code.
- Drawbacks of not using it: higher chance of errors when working on big projects due to misunderstandings. Also makes it hard for other programmers to collaborate and understand code.