

2321031

Trung Doan

Course Object Oriented Programming

Metropolia

Week4 : AI-Assisted Programming

Task1:

```
public class Calculator {
    private int sum;

    public Calculator() {
        this.sum = 0;
    }

    // Method to reset the calculator to zero
    public void reset() {
        this.sum = 0;
    }

    // Method to add an integer to the calculator
    // throw exception when add negative number
    public void add(int number) {
        if (number < 0) {
            throw new IllegalArgumentException("Negative numbers are not allowed");
        }
        this.sum += number;
    }

    // Method to return the current value of the calculator
    public int getValue() {
        return this.sum;
    }

    public static void main(String[] args) {
        Calculator calculator = new Calculator();
        calculator.add(5);
        calculator.add(10);
        System.out.println("Current sum: " + calculator.getValue()); // Output: Current sum: 15
        calculator.reset();
        System.out.println("After reset: " + calculator.getValue()); // Output: After reset: 0

        try {
            calculator.add(-5); // This will throw an exception
        } catch (IllegalArgumentException e) {
            System.err.println(e.getMessage()); // Output: Negative numbers are not allowed
        }
    }
}
```

## How to use AI:

- First I will create a Calculator.java and write comment of the methods that I needs according to the task requirements.
- Second I will copy-paste all the requirement from the assignment into the Copilot and let it generate the answer
- Third, I will copy the result and run the file
- However, I notice that it missing the handling exception case that is written inside the requirement.
- I have expixitely written this requirement to copilot. Then Copilot generate the method that include raising exception:

```
- throw new IllegalArgumentException("Negative numbers are not allowed");
```

- It also adding the testing case inside main too:

```
try {  
    calculator.add(-5); // This will throw an exception  
} catch (IllegalArgumentException e) {  
    System.err.println(e.getMessage()); // Output: Negative numbers are not allowed  
}
```

From this, here is my opinions about the ChatGPT Copilot:

### Pros:

- It can easily generate the whole program, as long as the programmer list out all the required methods, in this case are: add(), getValue(), and reset()
- It also generate test case for the methods

### Cons:

- When there are a specific requirement such as raising exception, the chatGPT might overlook this requirement. It means it just generates the program that fulfill all the requirements without cover all the edge cases. Just a functional program
- As a developer, he needs to the whole program and makes sure that the programs covers all the edge cases or the small requirements. Otherwise, he will need to write them explicitly to Copilot then Copilot will generate a better version of the code.