Assignment 4.2

AI-Assisted Coding

Name: Khai Cao

Group: ICT22-SW

I. Task 1.

GitHub Copilot is an AI-powered code completion tool that assists developers by generating code suggestions based on the provided context, comments, and existing code snippets. It uses machine learning to understand patterns in code and assists in writing code by suggesting completions for method bodies, loops, conditionals, and more.

In order to complete this assignment, I followed the steps below:

1. I created a new Java project and created a new class called Calculator along with its variable and methods. GitHub Copilot then suggested how the methods would look like.
2. After that, I created the main method and created a new Calculator object and added a while loop that runs until the user inputs an empty string. GitHub Copilot then suggested the code for the while loop.
3. Then I adjusted the code according to the requirements and made some extra modifications to the code.
4. Finally, I let GitHub Copilot add some explanations to the code. After finishing, the code looks like down below.

```java
public class Calculator {
    private int value;

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

    public int getValue() {
        return this.value;
    }

    public void add(int number) {
        this.value += number;
    }

    public void reset() {
        this.value = 0;
    }

    Run | Debug
    public static void main(String[] args) {
        Calculator calculator = new Calculator();

        while (true) {
            System.out.println("Current value: " + calculator.getValue());
            System.out.println("Input 'reset' to reset value");
            System.out.println("Input nothing to exit");
            System.out.print("Input: ");

            String input = System.console().readLine();

            // If input is empty, break the loop
            if (input.equals("")) {
                break;
            }

            // If input is "reset", reset the value
            if (input.equals("reset")) {
                calculator.reset();
                continue;
            }

            try {
                int number = Integer.parseInt(input);

                // If number is negative, throw exception
                if (number < 0) {
                    throw new NumberFormatException();
                }
                else calculator.add(number);

            } catch (NumberFormatException e) {
                System.out.println("Invalid input!");
            }

            System.out.println();
        }
    }
}
```

II. Benefits and drawbacks.

GitHub Copilot has several benefits that can enhance the coding experience, but it also comes with certain drawbacks:

1. Benefits.
   - Boost productivity
   - Improve code quality
   - Improve learning and exploring experience
   - Provide assistment in complex tasks
   - Save time

2. Drawbacks
   - Dependency on provided code
   - Potential security risks
   - Overreliance and reduce understanding
   - License issues
   - Language and context limitations
   - Limited adaptation to unique requirements