Credit Name: CSE 2140 2nd Language Programming
Assignment Name: MathTutor Mastery Project

How has your program changed from planning to coding to now? Please explain?

1. **Planning:**
   - Initially, the concept was straightforward: generate two random numbers, select an operator, display the problem, get the user's input, and then check if the answer is correct.
   - I also thought about the need to handle basic arithmetic operations (+, -, *, /) and user interaction.

```
Random random = new Random();
Scanner scanner = new Scanner(System.in);

// Generates the two random numbers between 1-10
int num1 = random.nextInt(10) + 1;
int num2 = random.nextInt(10) + 1;

// The array of possible operators
char[] operators = {'+', '-', '*', '/'};
// Randomly selects an operator
char operator = operators[random.nextInt(4)];

// Initialize the correct answer variable
double correctAnswer = 0;
```

2. **During Coding:**
   - I realized that division needed special handling since it could result in decimals and required rounding to avoid precision issues. So I added rounding to two decimal places for division results.
   - I added a condition to check if the user's answer is close enough to the correct answer (especially for floating-point numbers in division), using a tolerance of `0.01`.

```
            case '/':
                // Handles division ensuring no division by zero
                correctAnswer = (double) num1 / num2;
                correctAnswer = Math.round(correctAnswer * 100.0) / 100.0;  // Round to 2 decimal places
                break;
        }

        // Prompts the user for an answer
        System.out.println("Solve the following problem: " + num1 + " " + operator + " " + num2);
        double userAnswer = scanner.nextDouble();

        // Check if the user's answer is correct
        if (Math.abs(userAnswer - correctAnswer) < 0.01) {
            System.out.println("Correct! Well done.");
        } else {
            System.out.println("Incorrect. The correct answer is: " + correctAnswer);
        }
```

3. **Final Code:**
   - The final version includes handling for all the different operators, rounding for division operations, and a simple way to check answers while also accounting for precision in floating-point operations. These changes made the program work smoothly overall and avoided the issue of floating-point inaccuracies in division which I had experienced.