Exam Prep: Unit Testing

Math Enforcer

Your task is to test the following JavaScript class:

JS Code

You are provided with an implementation of the **`mathEnforcer`** object:

mathEnforcer.js

```
class MathEnforcer {

    static addFive(num) {
        if (typeof(num) !== 'number') {
            return undefined;
        }
        return num + 5;
    },
    static subtractTen(num) {
        if (typeof(num) !== 'number') {
            return undefined;
        }
        return num - 10;
    },
    static sum (num1, num2) {
        if (typeof(num1) !== 'number' || typeof(num2) !== 'number') {
            return undefined;
        }
        return num1 + num2;
    }

    static verifyNum(param) {
    if (typeof param !== 'number') throw new TypeError('Argument must be
number');
  }
};
```

The methods should function correctly for **positive**, **negative** and **floating-point** numbers. In case of **floating-point** numbers, the result should be considered correct if it is **within 0.01** of the correct value.

Hints

- Test how the program behaves when passing in **negative** values.
- Test the program with floating-point numbers (use Chai's **`closeTo()`** method to compare floating-point numbers).

What to submit?

Export the object in **mathEnforcer.js** and import it in your test file to test it. Submit a **zip** file containing the **mathEnforcer.js** and **tests folder** containing the **mathEnforcer.test.js**. **DO NOT!** include the **node_modules** folder.

File Name: MATH-ENFORCER.zip