Objective: The objective of this assignment is to test the understanding of the "call", "bind", and "apply" methods in JavaScript.

Instructions:

- Create a class named "Calculator" with the following methods:

  a. "add" method that accepts two numbers and returns their sum.

  b. "subtract" method that accepts two numbers and returns their difference.

  c. "multiply" method that accepts two numbers and returns their product.

  d. "divide" method that accepts two numbers and returns their quotient.

- Create a class named "ScientificCalculator" that extends the Calculator class and adds the following methods:

  a. "square" method that accepts a number and returns its square.

  b. "cube" method that accepts a number and returns its cube.

  c. "power" method that accepts two numbers and returns the first number raised to the power of the second number.

- Create an instance of the ScientificCalculator class
- **Using the "call" method,** invoke the "add" method of the Calculator class with arguments 10 and 5.
- **Using the "apply" method**, invoke the "subtract" method of the Calculator class with arguments 10 and 5.
- **Using the "bind" method,** create a new method named "multiplyByTwo" that multiplies a number by 2 and returns the result. Bind the "multiplyByTwo" method to the instance of the ScientificCalculator class.
- **Using the "bind" method,** create a new method named "powerOfThree" that raises a number to the power of 3 and returns the result. Bind the "powerOfThree" method to the instance of the ScientificCalculator class.
- **Optional:** Call the "multiplyByTwo" method on the instance of the ScientificCalculator class with argument 5 and print the returned result to the console.
- **Optional:** Call the "powerOfThree" method on the instance of the ScientificCalculator class with argument 2 and print the returned result to the console.

Desired Coding Practices:

1. Code should be indented properly.
2. Code should be readable.
3. Code should handle any edge cases foreseen.
4. Add comments wherever required.
5. Follow meaningful naming conventions, avoid generic naming conventions such as a,b,c,x,y,z etc.