**Part 1**

In this task we provide you with the start of a definition for a Shape class. It has three properties: name, sides, and sideLength. This class only models shapes for which all sides are the same length, like a square or an equilateral triangle.

We'd like you to:

* Add a constructor to this class. The constructor takes arguments for the name, sides, and sideLength properties, and initializes them.
* Add a new method calcPerimeter() method to the class, which calculates its perimeter (the length of the shape's outer edge) and logs the result to the console.
* Create a new instance of the Shape class called square. Give it a name of square, 4 sides, and a sideLength of 5.
* Call your calcPerimeter() method on the instance, to see whether it logs the calculation result to the browser's console as expected.
* Create a new instance of Shape called triangle, with a name of triangle, 3 sides and a sideLength of 3.
* Call triangle.calcPerimeter() to check that it works OK.

Try updating the live code below to recreate the finished example:

class Shape {

name;

sides;

sideLength;

}

Part 2

Next we'd like you to create a Square class that inherits from Shape, and adds a calcArea() method that calculates the square's area. Also set up the constructor so that the name property of Square object instances is automatically set to square, and the sides property is automatically set to 4. When invoking the constructor, you should therefore just need to provide the sideLength property.

Create an instance of the Square class called square with appropriate property values, and call its calcPerimeter() and calcArea() methods to show that it works OK.

Part 3

Create class Triple:

Has static field customName with value “Tripler”

Has static field description with value "I triple any number you provide"

Create static method calculate with n default arguments = 1 and return n \* 3

Create SquaredTriple subclass from Triple

Has static field longDescription

Has static field description with value “square the triple of any number you provide”

Override static method calculate return calculate(n) \* calculate(n) from parent class

Example test

console.log(Triple.description); // 'I triple any number you provide'

console.log(Triple.calculate()); // 3

console.log(Triple.calculate(6)); // 18

console.log(SquaredTriple.calculate(3)); // 81 (not affected by parent's instantiation)

console.log(SquaredTriple.description); // 'I square the triple of any number you provide'

console.log(SquaredTriple.longDescription); // undefined

console.log(SquaredTriple.customName); /