Note:

-Using Ecma6 Classes and Modules ,create Each Class at it's Own Module

-Link Module with each other with import and export

-Main module with Array Oof each class Object from same Type

PartOne:

1- Create Shape Base Abstract Class which contains color property as private with set and get,PrintColor and CalcArea and Calcperimeter methods which will return Zero in Shape Base.

2- Create Rectangle Class Which inherits from Shape Abstract Class

define Width and Height Properties for Rect Class as private with set and get

Not with zero or minus value print validation message to user

3- Creat Square Class Which inherits from Rect Class - override CalcArea , calcperimeter , printColor , toString which will display color , area and perimeter in rect and square classes

create array of Shapes which will contains set of objects from rect and square classes then display it's areas

4-create Circle Class with private fields (radius and x,y) with set and get properies

-Circle class inherit from Shape class with override toString

5- Create static property and static method inside Rect and Square classes to get number of objects created from rect and square Types

Part Two:

Create A Car Class has a Serial ,Name and a Speed property As private .

The SerialID for car create Unique random Number for each car

The Speed property is the Current Speed of the Car in Km/h

2- Implement an 'accelerate' method will increase the car's speed by 10,and log then new speed to console;

3-Implement a 'brake' method that will decrease the car's speed by 5,and log the new speed to the console;

4-Implement a static member represent the number of create Car ,print the serial number of car and total number of it inside static

Create 2 car objects and experiment with calling 'accelerate' and 'brake' multiple times on each of them.

DATA Car1 : 'BMW' going at 120 km/h

DATA CARA2: 'Mercedes' going at 95 km/h

Part Three:

1-Create Electric Car (Called EV) as a **CHILD** "class" of Car Besides a Name and Current Speed ,the EV also has the Current battery charge in % ('charge' property)

2-Implement a 'chargeBattery' method which takes an arguments 'chargeTo' and sets the battery charge to this value;

3-Implment an 'accelerate' method that will increase the car's speed by 20, and decrease the charge by 1% ,then log a message like this :

'Tesla going at 149 km/h, with a charge of 22%';

4- Create an electric car object and experiment with calling 'acceleracte ','brake' and 'chargeBattery'

(charge to 90%). Notice what happens when you ' accelerate

DATA CAR 1 :' Tesla' going at 120 km/h , with a charge of 23%

GOOD LUCK 😊