**EXSM 3936: JavaScript II - Objects & JSON**

Deadline: Sunday May 8, 2022 at 11:59 PM

<https://classroom.github.com/a/e_3m1UCC>

Introduction

This assignment is intended to cement your understanding of object oriented programming.

The most recent commit on the main branch will be marked.

Instructions

You are to write a class that will simulate a car. The class should be structured as follows (data types are intended, they do not need to be enforced):

1. *method:* constructor(make [string], model [string], engineCylCount [number], engineCylConfig [string], transmissionType [string])
2. *property:* make [string]
   1. Initialized from constructor parameter.
3. *property:* model [string]
   1. Initialized from constructor parameter.
4. *property:* engine [object]
   1. *property*: cylCount [number]
      * Initialized from constructor parameter.
   2. *property*: cylConfig [string]
      * Initialized from constructor parameter.
   3. *property:* running [boolean]
      * Initialized to false.
5. *property:* transmission [object]
   1. *property:* type [string]
      * Initialized from constructor parameter.
6. *property:* odometerKM [number]
   1. Initialized to 0.
7. *method:* startEngine()
   1. Sets the engine’s running property to true, if it is false.
   2. If the engine’s running property is already true, throw an exception.
8. *method:* stopEngine()
   1. Sets the engine’s running property to false, if it is true.
   2. If the engine’s running property is already false, do nothing.
9. *method:* drive(distance [number])
   1. If the engine is currently not running, throw an exception.
   2. Increase the odometerKM by a value equal to the distance parameter.

Once your class is constructed, **using the console template in your repository**, create a script that will (safely, remember to catch potential exceptions):

1. Instantiate a car.
2. Turn its engine on.
3. Drive for 100km.
4. Drive for 50km.
5. Turn the engine off.
6. Output the odometer reading to the console.

Criteria

| Requirement | Marks Available | Mark(s) Awarded | Comments |
| --- | --- | --- | --- |
| Script displays the value expected, and the steps outlined are followed. | 1 |  |  |
| Create New Car Profile: |  |  |  |
| The car constructor correctly initializes all basic properties. | 1 |  |  |
| The car constructor correctly initializes the child object properties (engine and transmission). | 2 |  |  |
| The behaviour of startEngine() is as outlined. | 0.5 |  |  |
| The behaviour of stopEngine() is as outlined. | 0.5 |  |  |
| The behaviour of drive() is as outlined. | 1 |  |  |
| Deductions: |  |  |  |
| README.md file is not present in the docs directory, or does not contain a reasonable minimum amount of information. | -1 |  |  |
| Version control is not used appropriately (-1 for vague or nonsensical commit messages, up to -2 for insufficient commits, -3 for a single commit regardless of message). | -3 |  |  |
| Whitespace is not managed (tab stops / formatting). | -1 |  |  |
| Variable, class, constant, property and/or method names do not meet [the code style requirements](https://docs.google.com/document/d/12iEtYtYiKwRAjhLayI-JVNVqXkud3X_PtOE9D0sU9Cw/edit) (-0.5 each declaration). | -3 |  |  |
| Missing semicolons (-0.5 each). | -2 |  |  |
| Unhandled exception(s) generated (-1 each scenario that generates one). | -3 |  |  |

**Total: /6**