

## CPSC1520 – JavaScript 8 Exercise: Classes and Objects

### Introduction

Classes and object oriented programming is an essential of your tool box as a developer. In this assessment we're going to set up a class in a file, export it and use it in our index JavaScript file. We're going to create a fundamental quick racing app for fun, to understand the fundamentals of classes and objects.

### Exercise Step 1: Create the Class and use it in your index.js

1. Download and extract the package given. Install the packages needed using "npm install" like we've done in class.
  - a. Once your packages are installed (you should be able to see the node\_modules folder) you can run the code by using "npm run start".
2. Open your "car.js" file in your folder, create a class named "Car" and export it, it should like below.

```
import { createCarElement } from '../dom/car.js'

class Car {

}

export { Car }
```

3. Import the class "Car" into your index.js file so that you can use this class. We'll be using this in later steps that you can use.
4. In your newly created "Car" class create a constructor that is going to take one parameter called "name" and add the following lines to the constructor:

```
this.name = name
this.element = createCarElement(this.name)
```

Observe here that we've created two attributes on the class "name" and "element" that we can use in other methods.

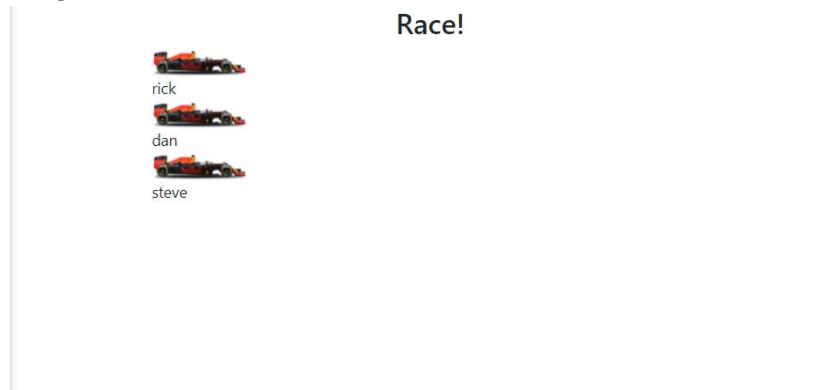
5. In your car class add a new method named "addToRace" this will take one parameter named "raceElement". In the "addToRace" method add the following lines of code.

```
this.speed = Math.random()*2
this.distance = 0
raceElement.append(this.element)
```

Observe here that we have created two new attributes "speed" and "distance", we'll use these

in other methods. We also append the attribute “element” created in the previous step to the raceElement parameter that we are going to pass in when we create call this function.

6. In your index.js file in the map (where you’re looping over all of the racers) create a new variable named “car” and create a new “Car” object.
  - a. After you have created your new car object. Call the method “addToRace” with the “race” variable so that you can add the cars to the page.
  - b. Once you have completed the above, once you refresh your page it should like the image below.



7. The last step here is that we want to make our cars race! We’re going to do that by creating a method named “drive” in our car class that’s going to take no parameters.

To simulate our racing we’re going to add the following lines in our “drive” method.

```
let s = setInterval(()=> {  
  this.distance += this.speed  
  this.element.style.marginLeft = this.distance + 'px'  
  if (this.distance > 500) {  
    clearInterval(s)  
  }  
}, 10)
```

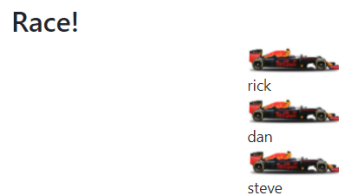
Notice that we’re continuously adding the speed to the distance, since we’re calling this every 10ms in the “setInterval” call back. We’re also stopping the cars by clearing the interval once the distance is greater than 500, if you want them to race a larger distance increase this or use percentages (this last part is totally optional).

8. The last step is in the index.js want to call our “drive” method after we’ve added cars to the race using the “addToRace” method.

- a. When the pulse pounding race is in action it should look like below.



- b. Once race is complete the cars should be on the right like below.



9. Once you've got this working like the above (using classes and objects of course) you can go to the following step.

### Exercise Step 2: Upload and Zip your site to moodle.

1. Once Complete, stop the parcel process, and delete the parcel-cache folders and node\_modules folders.
2. Zip up your project and ensure that your entire project is included.
3. Submit that to moodle.

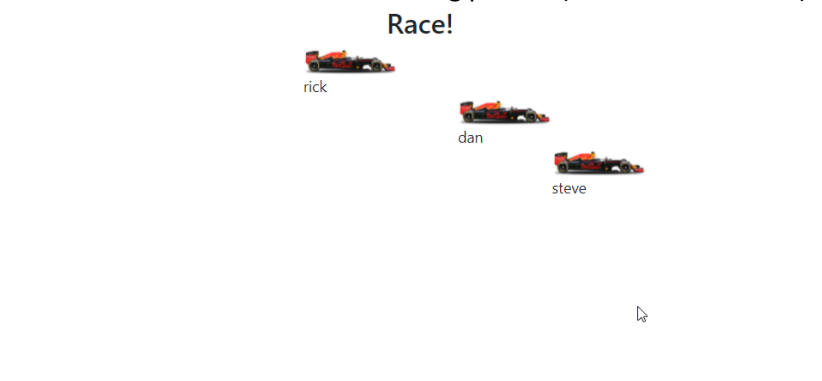
Note if it's too large then you probably have included your node\_modules (you'll get no marks if they're included anyways because you read the Grading section below so you'll delete it.)

### Grading

I'll give full marks if:

- There is NO node\_modules and NO .parcel\_cache folders included in the project.
- Everything is imported correctly.
- The car class is correct which includes the constructor, the drive and the addToRace methods.

- The end result looks like the following picture (when it's in motion)



If you don't follow the instructions you'll get a zero. There are no marks in between.