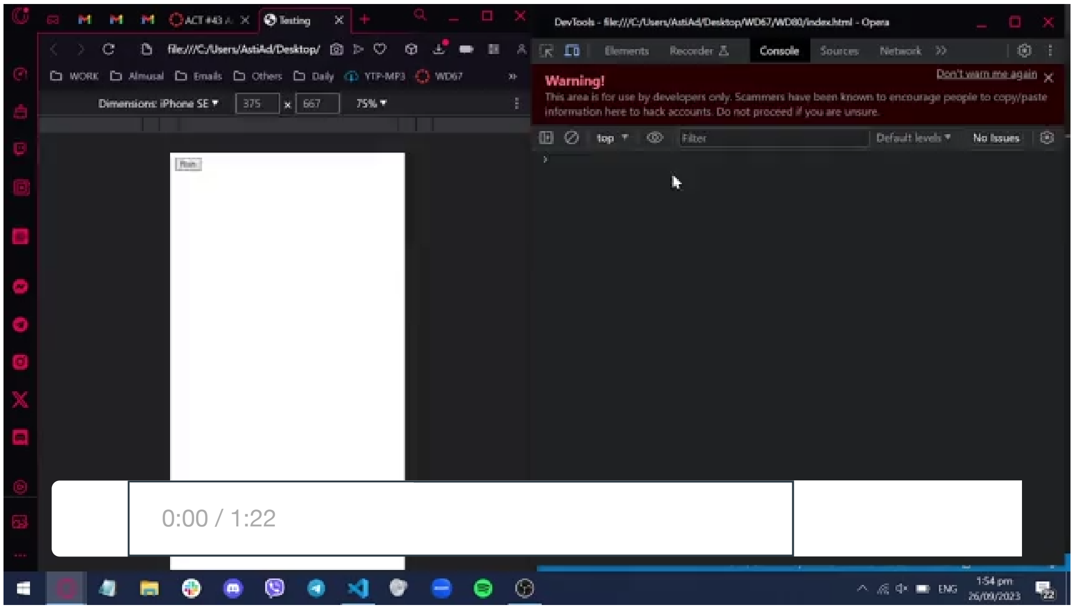
ACT #41 Virtual Car

New Attempt

Due No Due Date **Points** 70 **Submitting** a file upload **File Types** js

SOFT DEADLINE: 12/7/2023 11:15 AM

HARD DEADLINE: 12/7/2023 11:20 AM



Warning!
This area is for use by developers only. Scammers have been known to encourage people to copy/paste information here to hack accounts. Do not proceed if you are unsure.

Instruction

Create a program that recreates a car.

Tools

JavaScript, HTML, Visual Studio Code

Description

- **[Follow the Submit Your Work steps. (https://kodego.instructure.com/courses/379/pages/7-dot-1-essential-javascript-functions?module_item_id=17474)](https://kodego.instructure.com/courses/379/pages/7-dot-1-essential-javascript-functions?module_item_id=17474)**
- Name the file **garage.js**
- Write a script that does the following. The script would:
  - Create a class called `Car`.

  - This class would have the following properties:
    - `name` - no default (parameter)
    - `manufacturer` - no default (parameter)
    - `acceleration` - no default (parameter)
    - `speed` - default 0
  - This class would have the following methods:
    - `start`
      - Starts the car.
      - Sets the `speed` to 30.
      - This method can only be used if the car hasn't started yet.
      - Display through the console that the car has started in the format `[car name] has started! Speed at 30`
    - `accelerate`
      - Adds the car's `acceleration` property to the current speed.
      - This method can only be used **once the car has started**.
      - Display through the console the new `speed` in the format `[car name] has accelerated! New speed: 130`
    - `decelerate`
      - Halves the current speed of the car.
        - The speed can only go down to a minimum of 1 when using this method.
      - This method can only be used **once the car has started**.
      - Display through the console the new `speed` in the format `[car name] has decelerated! New speed: 65`
    - `checkSpeed`
      - Displays through the console the current `speed`.
      - Can be used even if the car hasn't started yet in the format `Current speed: 65`
    - `stop`
      - Stops the car.
      - You can only stop the car if the car has started.
      - Set the `speed` to 0.
      - The car would need to be started again to accelerate/decelerate.
      - Display through the console that the car has stopped in the format `[car name] has stopped.`
- Create a `Car` object, asking the user for the name, manufacturer, and acceleration.
  - The name, manufacturer, and acceleration would then be saved as the `Car` object's `name`, `manufacturer`, and `acceleration`, respectively.
- Set up a loop that asks the user which method they want to run with the created Car instance (similar to the previous fruit store activity).
  - Use numbers to designate the functions (i.e. 1. start, 2. accelerate, 3. decelerate, 4. check speed, 5. stop, 6. end program)

- Submit your JavaScript file only here.

Validation

- Aside from the requirements stated above, the program should also display an error message in the case of...
  - The user setting the `Car` object acceleration to zero or negative number.

Notes and Tips

- You can use `speed` to determine whether the car has started on not.

Started Code

```
function runActivity() {
  class Car {
    constructor(/* check the properties you need to add*/) {
      //set your properties using the this keyword
      this.speed = 0;
    }

    //all of these interact with the speed property in some way
    //no need for parameters
    //no need for return
    start() {}

    accelerate() {}

    decelerate() {}

    checkSpeed() {}

    stop() {}
  }

  //you may need to add more here
  let name = prompt("Give me the car's name.");
  let manufacturer = prompt("Give me the car's manufacturer.");
  let acceleration = prompt("Give me the car's acceleration.");
  let myCar = new Car(/* fill up the arguments */);

  let i = 0;
  while (i == 0) {
    let choice = Number(
      prompt(
        "(1) Start (2) Accelerate (3) Decelerate (4) Check Speed (5) Stop (6) End program"
      )
    );
    switch (
      choice
      /* have each number correspond to a method */
    ) {
    }
  }
}
```

Test Cases

Input	Result

Prompt order:
Comet
Toyota
20.5
1
2
3
4
5
6
	`Comet has started! Speed at 30.`
	`Comet has accelerated! New speed: 50.5`
	`Comet has decelerated! New speed: 25.25`
	`Current speed: 25.25`
	`Comet has stopped.`

Prompt order:
Toothache
Beetle
10
3
2
4
1
3
3
3
3
3
	`Toothache has not started yet.`
	`Toothache has not started yet.`
	`Current speed: 0`
	`Toothache has started! Speed at 30.`
	`Toothache has decelerated! New speed: 15`
	`Toothache has decelerated! New speed: 7.5`
	`Toothache has decelerated! New speed: 3.75`
	`Toothache has decelerated! New speed: 1.875`
	`Toothache has reached it's minimum speed.`
	`Toothache has stopped.`

5 6	