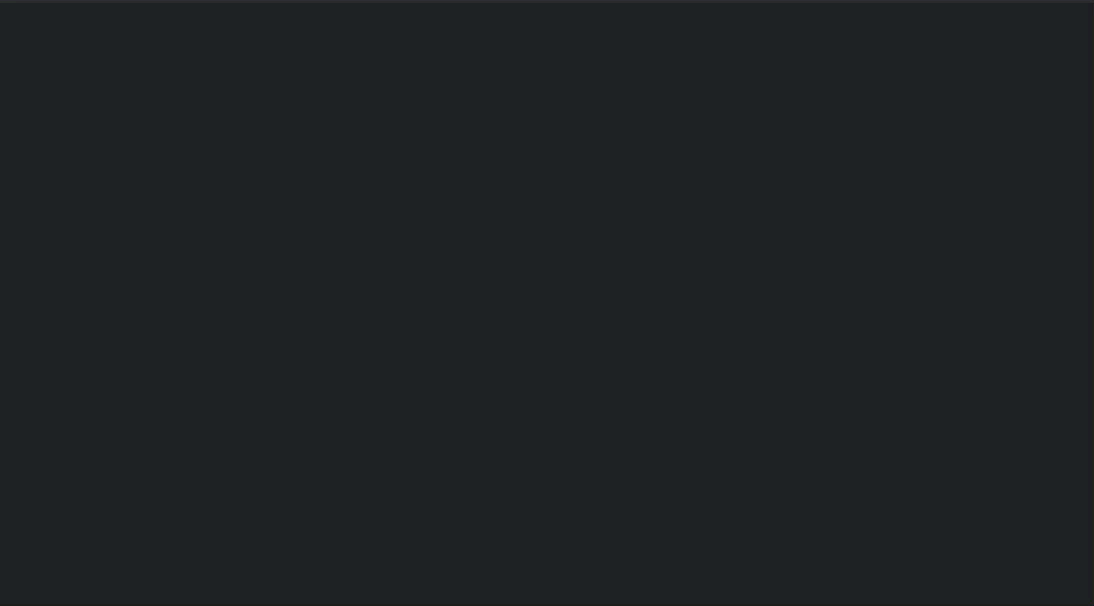
**Assignment Title:** Vehicle Race Simulation

Sample runs of target program (gif): [Linked here](https://eriecotech-my.sharepoint.com/:i:/g/personal/jklins_ects_org/EXC_ZWExwDRLsejto19r6nwBKyUd6-xM2D0II5F4Kl-EcQ?e=z8V7su) if it doesn’t load.

****

**Requirements:**

1. **Class Creation:**
   * Create two classes: **VehicleRace** and **Vehicle**.
   * The **Vehicle** class should have private attributes for the brand, speed, and distance traveled, as well as appropriate constructors, getters, and a method to update the distance the car has traveled. **See pseudocode below**.
   * The **VehicleRace** class should contain the main method and the logic for the race simulation. **See pseudocode below.**
2. **Initialization:**
   * In the **VehicleRace** class, create two vehicle objects with initial speeds within the range specified (95 to 105) and display their initial information.
3. **Race Simulation:**
   * Simulate the race until one of the cars reaches the finish line, which is set to 500 miles.
   * Inside a loop, update the distances traveled by each car in each race step.
   * Randomly vary the speeds of the cars in each step within a range of -10 to +10 mph.
   * Pause the program for one second (1000 milliseconds) after each race step to visualize the race progress.
4. **Display Information:**
   * Display the brand, current speed, and distance traveled of each car in each race step.
5. **Determining the Winner:**
   * Determine the winner based on which car traveled a distance greater than or equal to 500 miles first. If both pass 500 miles on the same iteration, the winner is the one that traveled the furthest. If they both traveled the same distance, declare it a tie.
6. **Output:**
   * Display a message indicating the winner of the race (or a tie).

# Pseudocode:

**VehicleRace Class**

Constants:

INITIAL\_SPEED\_MIN = 105

INITIAL\_SPEED\_MAX = 110

DISTANCE\_TO\_FINISH = 500

MILLIDELAY = 1000

Main Method:

Create Vehicle object car1 with a random speed in the range (INITIAL\_SPEED\_MIN, INITIAL\_SPEED\_MAX)

Create Vehicle object car2 with a random speed in the range (INITIAL\_SPEED\_MIN, INITIAL\_SPEED\_MAX)

Display "Initial Information:"

Display car1's brand, speed, and distance traveled

Display car2's brand, speed, and distance traveled

Display "LET THE RACE BEGIN!"

While car1's distance traveled < DISTANCE\_TO\_FINISH and car2's distance traveled < DISTANCE\_TO\_FINISH:

Call raceCars method with car1 and car2 as parameters

Pause the program for MILLIDELAY milliseconds

Determine the winner based on which car traveled a distance greater than or equal to DISTANCE\_TO\_FINISH. If both passed the finish line after the same iteration, the winner is the one that traveled the furthest. If they both traveled the same distance, declare it is a tie.

Display the winner's brand with "won the race!"

getRandomSpeed Method:

Generate a random speed within the range (INITIAL\_SPEED\_MIN, INITIAL\_SPEED\_MAX)

Return the random speed

raceCars Method with parameters car1 and car2:

speed1 = car1's speed

speed2 = car2's speed

Randomly vary speed1 and speed2 within the range (-5 to +5)

Update car1's distance traveled by adding speed1 to it

Update car2's distance traveled by adding speed2 to it

Display car1's brand, speed1, and updated distance traveled

Display car2's brand, speed2, and updated distance traveled

**Vehicle Class**

Attributes:

brand

speed

distanceTraveled

Constructor with parameters brand and speed:

Initialize brand with the provided value

Initialize speed with the provided value

Initialize distanceTraveled to 0

Getter methods for brand, speed, and distanceTraveled

updateDistance Method with parameter speed:

Update distanceTraveled by adding speed to it

Paste your vehicle class here:

|  |
| --- |
|  |

Paste your vehicleRace class here:

|  |
| --- |
|  |

Paste an image of your output here:

|  |
| --- |
|  |