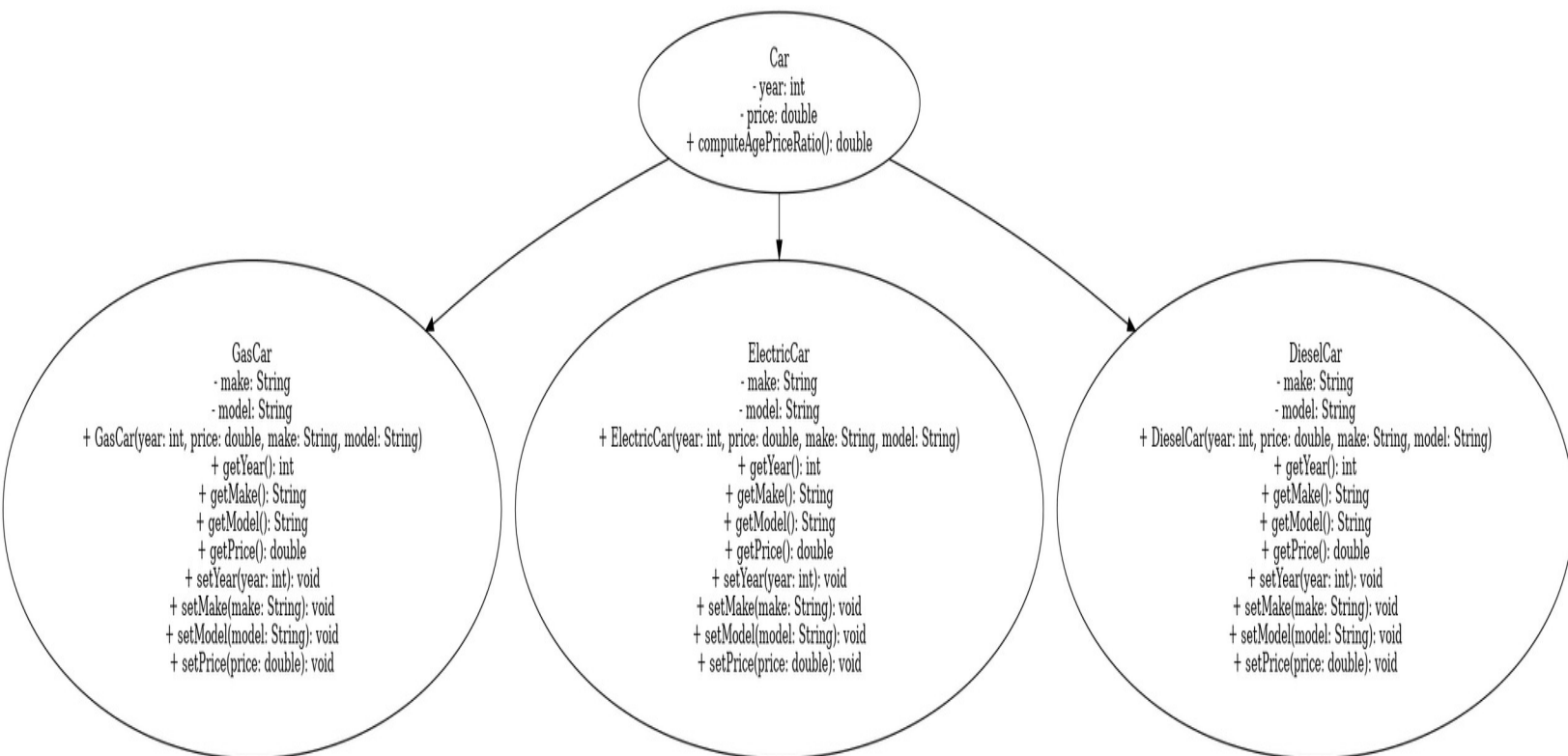


Lab 2

Name: Jacob Archer

RedID: 824816519



QUESTION 1

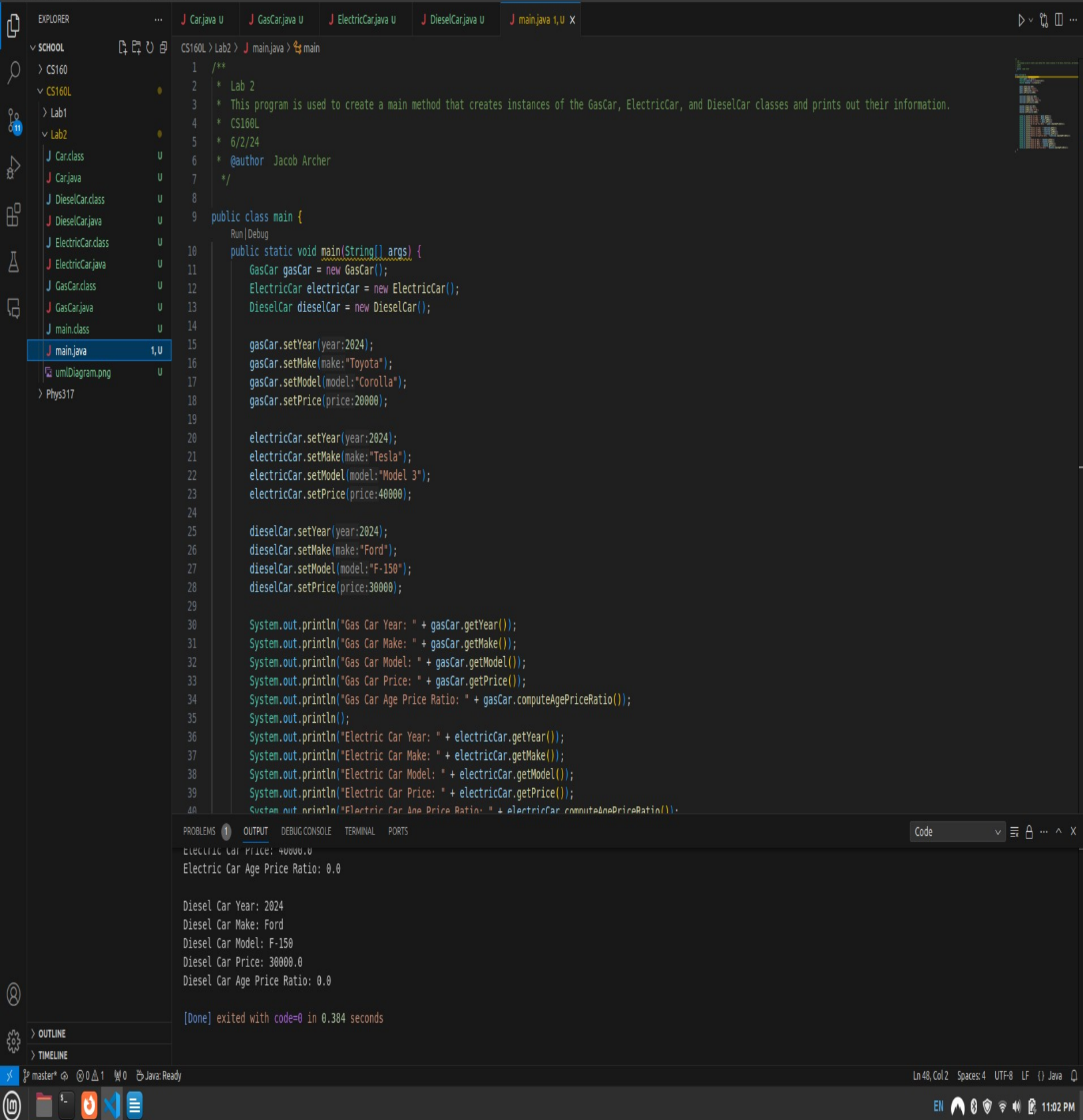
Why create separate classes for different types of objects?

- Creating separate classes allows for better organization and maintainability of code.
- Each class can encapsulate the unique properties and behaviors of its corresponding object type.
- Specific classes make the code more modular and easier to understand, promoting code reusability and simplifying maintenance.

QUESTION 2

Difference between default constructor and parameterized constructor:

- A default constructor is provided by the programming language if no constructor is explicitly defined in the class, initializing the object with default values.
- A parameterized constructor accepts parameters during object instantiation, allowing for customized initialization of object properties with specific values provided by the caller.
- While a default constructor initializes with default values, a parameterized constructor provides flexibility for tailored object construction.



CS160L > Lab2 > J main.java > main

```
1 /**
2  * Lab 2
3  * This program is used to create a main method that creates instances of the GasCar, ElectricCar, and DieselCar classes and prints out their information.
4  * CS160L
5  * 6/2/24
6  * @author Jacob Archer
7  */
8
9 public class main {
10     Run | Debug
11     public static void main(String[] args) {
12         GasCar gasCar = new GasCar();
13         ElectricCar electricCar = new ElectricCar();
14         DieselCar dieselCar = new DieselCar();
15
16         gasCar.setYear(year:2024);
17         gasCar.setMake(make:"Toyota");
18         gasCar.setModel(model:"Corolla");
19         gasCar.setPrice(price:20000);
20
21         electricCar.setYear(year:2024);
22         electricCar.setMake(make:"Tesla");
23         electricCar.setModel(model:"Model 3");
24         electricCar.setPrice(price:40000);
25
26         dieselCar.setYear(year:2024);
27         dieselCar.setMake(make:"Ford");
28         dieselCar.setModel(model:"F-150");
29         dieselCar.setPrice(price:30000);
30
31         System.out.println("Gas Car Year: " + gasCar.getYear());
32         System.out.println("Gas Car Make: " + gasCar.getMake());
33         System.out.println("Gas Car Model: " + gasCar.getModel());
34         System.out.println("Gas Car Price: " + gasCar.getPrice());
35         System.out.println("Gas Car Age Price Ratio: " + gasCar.computeAgePriceRatio());
36         System.out.println();
37         System.out.println("Electric Car Year: " + electricCar.getYear());
38         System.out.println("Electric Car Make: " + electricCar.getMake());
39         System.out.println("Electric Car Model: " + electricCar.getModel());
40         System.out.println("Electric Car Price: " + electricCar.getPrice());
41         System.out.println("Electric Car Age Price Ratio: " + electricCar.computeAgePriceRatio());
42     }
43 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

Electric Car Price: 40000.0
Electric Car Age Price Ratio: 0.0

Diesel Car Year: 2024
Diesel Car Make: Ford
Diesel Car Model: F-150
Diesel Car Price: 30000.0
Diesel Car Age Price Ratio: 0.0

[Done] exited with code=0 in 0.384 seconds

master* 0 1 0 Java: Ready

Ln 48, Col 2 Spaces: 4 UTF-8 LF {} Java

EN 11:02 PM

