Cabenta, Joseph S.
BSCS – C204

October 18, 2025
7OOP

FINALS LAB TASK #1

I. PROBLEM

For this program, you are tasked to define the following:

Class - Car:

- Properties:
  - `color` (type: str): Represents the color of the car.
  - `price` (type: float): Holds the price of the car.
  - `size` (type: str): Indicates the size of the car, where 'S' represents small, 'M' represents medium, and 'L' represents large.
- Constructor:
  - `__init__(self, color: str, price: float, size: str)`: Initializes the car's `color`, `price`, and `size` properties. The `size` is standardized to uppercase using `size.upper()`.
- Methods
  - Getter Methods:
    - `get_color(self) -> str`: Returns the car's color.
    - `get_price(self) -> float`: Returns the car's price.
    - `get_size(self) -> str`: Returns the car's size.
  - Setter Methods:
    - `set_color(self, color: str) -> None`: Sets the car's color to the specified value.
    - `set_price(self, price: float) -> None`: Sets the car's price to the specified value.
    - `set_size(self, size: str) -> None`: Sets the car's size to the specified value. The size should be one of 'S' for small, 'M' for medium, or 'L' for large. Use conversion of lowercase characters to uppercase using `size.upper()`.
  - `__str__` Method:
    - `__str__(self) -> str`: Returns a formatted string representing the car, following the format "Car (color) - P(price, formatted to two decimal places) - (size descriptor)". The size descriptor is determined based on the size character ('small' for 'S', 'medium' for 'M', and 'large' for 'L').
    - Example Strings:
      - For a red car priced at 19999.85 and of medium size: "Car (red) - P19999.85 - medium"
      - For a blue car priced at 50000.00 and large: "Car (blue) - P50000.00 - large"

II. CODE

```python
class Car:
    def __init__(self, color: str, price: float, size: str):
        self.__color = color
        self.__price = price
        self.__size = size.upper()

    # Setter
    def set_color(self, color: str) -> None:
        self.__color = color

    def set_price(self, price: float) -> None:
        self.__price = price

    def set_size(self, siz: str) -> None:
        self.__size == size.upper()

    # Getter

    def get_color(self) -> str:
        return self.__color

    def get_price(self) -> float:
        return self.__price

    def get_size(self) -> str:
        return self.__size

    # String Representaion
    def __str__(self) -> str:
        if self.__size == "S":
            size_desc = "small"
        elif self.__size == "M":
            size_desc = "medium"
        elif self.__size == "L":
            size_desc = "large"
        else:
            size_desc = "unknown"
        return f"Car ({self.__color}) - P{self.__price: .2f} - {size_desc}"

    # Main
if __name__ == "__main__":
    # Sample Input 1
    car1 = Car( color: "red",  price: 19999.85,  size: "M")
    print(car1)

    # Sample Input 2
    car2 = Car( color: "blue",  price: 50000.00,  size: "L")
    print(car2)

    # Sample Input 3
    car3 = Car( color: "green",  price: 12345.67,  size: "S")
    print(car3)
```

III. SAMPLE OUTPUT