Zabala, Rhaldynyl Brian F.
C203

**Encapsulation**

**Code:**

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

    #getters
    def get_color(self):
        return self.__color

    def get_price(self):
        return self.__price

    def get_size(self):
        return self.__size

    #setters
    def set_color(self, c: str):
        self.__color = c

    def set_price(self, p: float):
        self.__price = p

    def set_size(self, s: str):
        self.__size = s.upper()

    def __str__(self):
        if self.__size == 'S':
          size_desc = 'small'
        elif self.__size == 'M':
          size_desc = 'medium'
        elif self.__size == 'L':
          size_desc = 'large'
        else:
          size_desc = 'Invalid'

        return f"Car ({self.__color}) - ₱{self.__price:.2f} - {size_desc}"


print('Action: Invoking the Car class constructor using Car("red", 19999.85, \'M\').')
car1 = Car("red", 19999.85, 'M')
print("Output:")
print(car1)
print()

print('Action: Invoking the Car class constructor using Car("blue", 50000.00, \'L\').')
car2 = Car("blue", 50000.00, 'L')
print("Output:")
print(car2)
print()

print('Action: Invoking the Car class constructor using Car("green", 12345.67, \'S\').')
car3 = Car("green", 12345.67, 'S')
print("Output:")
print(car3)
```

**Output:**

```
Ln: 1,  Col: 1
▶ Run    ↱ Share    $    Command Line Arguments

Action: Invoking the Car class constructor using Car("red", 19999.85, 'M').
Output:
Car (red) - 19999.85 - medium

Action: Invoking the Car class constructor using Car("blue", 50000.00, 'L').
Output:
Car (blue) - 50000.00 - large

Action: Invoking the Car class constructor using Car("green", 12345.67, 'S').
Output:
Car (green) - 12345.67 - small


** Process exited - Return Code: 0 **
```