

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

```
1  class Car:
2      def __init__(self, color: str, price: float, size: str):
3          self.__color = color
4          self.__price = price
5          self.__size = size.upper()
6
7          # Setter
8      def set_color(self, color: str) -> None:
9          self.__color = color
10
11     def set_price(self, price: float) -> None:
12         self.__price = price
13
14     def set_size(self, size: str) -> None:
15         self.__size == size.upper()
16
17     # Getter
18
19     def get_color(self) -> str:
20         return self.__color
21
22     def get_price(self) -> float:
23         return self.__price
24
25     def get_size(self) -> str:
26         return self.__size
27
28     # String Representation
29     def __str__(self) -> str:
30         if self.__size == "S":
31             size_desc = "small"
32         elif self.__size == "M":
33             size_desc = "medium"
34         elif self.__size == "L":
35             size_desc = "large"
36         else:
37             size_desc = "unknown"
38
39
40         return f"Car ({self.__color}) - P{self.__price: .2f} - {size_desc}"
41
42     # Main
43     if __name__ == "__main__":
44         # Sample Input 1
45         car1 = Car( color="red", price=19999.85, size="M")
46         print(car1)
47
48         # Sample Input 2
49         car2 = Car( color="blue", price=50000.00, size="L")
50         print(car2)
51
52         # Sample Input 3
53         car3 = Car( color="green", price=12345.67, size="S")
54         print(car3)
```

III. SAMPLE OUTPUT

```
C:\Users\COMLAB\PycharmProjects\pyt
Car (red) - P 19999.85 - medium
Car (blue) - P 50000.00 - large
Car (green) - P 12345.67 - small

Process finished with exit code 0
```