S.No: 26	Exp. Name: ***Display the car details***	Date:2025-04-23

ID: 24070521230

Aim:

Follow the instructions to create a Python program modeling cars with specific types: Car1 and Car2. Begin by defining a base class **Car** with attributes **brand**, **price**, **model,** and **color**. Subsequently, create two derived classes, **Car1** and **Car2**, both inheriting from the Car class. Each derived class should introduce its attributes, and:

- Implement a method **display_details** in the base class Car to print the common attributes (brand, price, model, color).
- Override the **display_details** method in each derived class (Car1 and Car2) to print the brand, price, model, and color respectively.

Input Format:

For Car1:

- The first line contains the brand, price, model, and color of the Car1, separated by spaces.

For Car2:

- The first line contains the brand, price, model, and color of the Car2, separated by spaces.

Output Format:

- The first four lines should display information about the Car1, including the brand, price, model, and color.
- The second four lines should display information about the Car2, including the brand, price, model, and color.

Note:

- Price must be a positive float.
- Refer to the displayed test cases for better understanding.
- For simplicity, code for reading inputs has already been provided.

Source Code:

carDetails.py

```
class Car:
   # write your code here...
   def __init__(self,brand,price,model,color):
      self.brand= brand
      self.price= price
      self.model = model
      self.color= color

   def display_details(self):
      print(f"{self.brand}")
      print(f"{self.price}")
      print(f"{self.model}")
      print(f"{self.color}")
def get_car_details():
   brand = input("brand: ")
   price = float(input("price: "))
   model = input("model: ")
   color = input("color: ")
   return brand, price, model, color
```

```
print("Details for Car 1:")
car1_brand, car1_price, car1_model, car1_color = get_car_details()
car1=Car(car1_brand,car1_price,car1_model,car1_color)
# create an object car1


print("Details for Car 2:")
car2_brand, car2_price, car2_model, car2_color = get_car_details()
car2 = Car(car2_brand,car2_price,car2_model,car2_color)
# Create the second car object


print("Car 1 Details:")
# Display details of the car1
car1.display_details()

print("Car 2 Details:")
# Display details of the car1
car2.display_details()
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Details for Car 1: Nano
brand: Nano
price: 120000
model: Magic
color: Yellow
Details for Car 2: Innova
brand: Innova
price: 200000
model: xu
color: White
Car 1 Details:
Nano
120000.0
Magic
Yellow
Car 2 Details:
Innova
200000.0
xu
White