Multilevel Inheritance with Constructor Chaining in C++

This program demonstrates multilevel inheritance in C++ using constructor chaining. It includes parameterized constructors, destructors, and method calls showing how constructor and destructor calls flow through the inheritance hierarchy: Vehicle → MotorVehicle → Car.

```
#include <iostream>
#include <string>
using namespace std;

// Base class
class Vehicle {
protected:
    string brand;
    int wheels;

public:
    Vehicle(string b, int w) {
        brand = b;
        wheels = w;
        cout << "Vehicle constructor called for brand: " << brand << endl;
    }

    void showVehicleInfo() {
        cout << "Brand: " << brand << ", Wheels: " << wheels << endl;
    }

    ~Vehicle() {
        cout << "Vehicle destructor called for brand: " << brand << endl;
    }
};

// Derived class 1
class MotorVehicle : public Vehicle {
protected:
    string engineType;

public:
    MotorVehicle(string b, int w, string e) : Vehicle(b, w) {
        engineType = e;
        cout << "MotorVehicle constructor called, Engine type: " << engineType << endl;
    }

    void showMotorInfo() {
        showVehicleInfo();
        cout << "Engine Type: " << engineType << endl;
    }

    ~MotorVehicle() {
        cout << "MotorVehicle destructor called for engine: " << engineType << endl;
    }
};

// Derived class 2
class Car : public MotorVehicle {
private:
    int seats;
    string fuelType;

public:
    Car(string b, int w, string e, int s, string f) : MotorVehicle(b, w, e) {
        seats = s;
        fuelType = f;
        cout << "Car constructor called. Seats: " << seats << ", Fuel: " << fuelType << endl;
    }

    void showCarInfo() {
        showMotorInfo();
        cout << "Seats: " << seats << ", Fuel Type: " << fuelType << endl;
    }
```

```
    ~Car() {
        cout << "Car destructor called for " << brand << endl;
    }
};

int main() {
    cout << "=== Creating Car Object ===" << endl;
    Car c("Toyota", 4, "Petrol Engine", 5, "Petrol");

    cout << "\n=== Displaying Car Details ===" << endl;
    c.showCarInfo();

    cout << "\n=== Program Ending (Destructors will be called) ===" << endl;
    return 0;
}

Expected Output:
-----------------
=== Creating Car Object ===
Vehicle constructor called for brand: Toyota
MotorVehicle constructor called, Engine type: Petrol Engine
Car constructor called. Seats: 5, Fuel: Petrol

=== Displaying Car Details ===
Brand: Toyota, Wheels: 4
Engine Type: Petrol Engine
Seats: 5, Fuel Type: Petrol

=== Program Ending (Destructors will be called) ===
Car destructor called for Toyota
MotorVehicle destructor called for engine: Petrol Engine
Vehicle destructor called for brand: Toyota
```