Scenario 1: Pointers to Class Members You're developing a vehicle management system with a class Vehicle that has several data members (for example, an integer speed and a float fuel Level and a member function displayStatus(). Parts of the system need to dynamically modify and display these attributes using pointers to class members.

(a) How would you declare a pointer to the speed data member and a pointer to the member function displayStatus()? (b) Provide a code snippet that uses these pointers to set the speed of a Vehicle object and then calls displayStatus(). (c) What are some potential pitfalls when using pointers to class members, and how can they be mitigated?

## (a) Declaring Pointers to Class Members

To declare a pointer to the `speed` data member:

int Vehicle::*ptrSpeed;

To declare a pointer to the `displayStatus` member function:
void (Vehicle::*ptrDisplay)() const;

## (b) Code Snippet

```
#include <iostream>

class Vehicle {
public:
   int speed;
   float fuelLevel;

   Vehicle(int s, float f) : speed(s), fuelLevel(f) {}

   void displayStatus() const {
      std::cout << "Speed: " << speed << " km/h, Fuel Level: " << fuelLevel << " L" <<
std::endl;
   }
};

int main() {
   Vehicle car(0, 50.0f);  // Create a vehicle instance

   // Pointer to data member
   int Vehicle::*ptrSpeed = &Vehicle::speed;
```

```
    // Pointer to member function
    void (Vehicle::*ptrDisplay)() const = &Vehicle::displayStatus;

    // Modify speed using pointer to member
    car.*ptrSpeed = 80;  // Access through object
    Vehicle* ptrCar = &car;
    ptrCar->*ptrSpeed = 100;  // Access through pointer

    // Call displayStatus() using pointer to member function
    (car.*ptrDisplay)();
    (ptrCar->*ptrDisplay)();

    return 0;
}
```

## (c) Potential Pitfalls and Mitigations

**1. Complex Syntax and Readability Issues**

```
using SpeedPtr = int Vehicle::*;
using DisplayFuncPtr = void (Vehicle::*)() const;
```

**2. Dangling Pointers**

**3. Null Pointers to Members**

```
if (ptrDisplay) {
    (car.*ptrDisplay)();
}
```

**4. Limited Use with Polymorphism**

**Problem: If `displayStatus()` is overridden in a derived class, calling it via a pointer-to-member may not exhibit polymorphic behavior unless called on an actual object of the derived class.**

**Solution: Consider using function pointers to virtual functions if polymorphism is needed.**