



6. Null Object - Design Pattern

Problem:

```
private static void printVehicleDetails(Vehicle vehicle){  
    System.out.println("Seating Capacity: " + vehicle.getSeatingCapacity());  
    System.out.println("Fuel Tank Capacity: " + vehicle.getTankCapacity());  
}
```

What's the issue here?

If the client calls this method using a null vehicle object, it will give a null pointer exception!!

Solution:

```
private static void printVehicleDetails(Vehicle vehicle) {  
    ✓ if (vehicle != null) {  
        System.out.println("Seating Capacity: " + vehicle.getSeatingCapacity());  
        System.out.println("Fuel Tank Capacity: " + vehicle.getTankCapacity());  
    }  
}
```

But it is not an optimal solution!

Because in real-world projects, there will be a large number of classes and methods.

Null Object DP!!!

- replace null with NULL Object
- It may do nothing or default behavior

Code

```
// Sachin Mahawar
#include <bits/stdc++.h>
using namespace std;

// Vehicle interface
class Vehicle
{
public:
    virtual int getSeatingCapacity() = 0;
    virtual int getFuelCapacity() = 0;
};

// Concrete Car class
class Car : public Vehicle
{
public:
    int getSeatingCapacity()
    {
        return 5;
    }
    int getFuelCapacity()
    {
        return 50;
    }
};

// Concrete Bus class
class Bus : public Vehicle
{
public:
    int getSeatingCapacity()
    {
        return 40;
    }
    int getFuelCapacity()
    {
        return 500;
    }
};

// Nullvehicle : default behaviour
class NullVehicle : public Vehicle
{
public:
    int getSeatingCapacity()
    {
        return 0;
    }
    int getFuelCapacity()
    {
        return 0;
    }
};

// Vehicle factory
class VehicleFactory
{
public:
    Vehicle *getVehicleObj(string vehicle)
```

```

    {
        if (vehicle == "Car")
            return new Car();
        else if (vehicle == "Bus")
            return new Bus();
        else
            return new NullVehicle();
    }
};

int main()
{
    VehicleFactory factory;
    Vehicle *myVehicle = factory.getVehicleObj("Car");

    cout << "Fuel: " << myVehicle->getFuelCapacity() << "\n";
    cout << "Seats: " << myVehicle->getSeatingCapacity() << "\n";

    // we have NullObj instead of null :)
    myVehicle = factory.getVehicleObj("Bike");

    // Here it will not throw error, instead shows default
    cout << "Fuel: " << myVehicle->getFuelCapacity() << "\n";
    cout << "Seats: " << myVehicle->getSeatingCapacity() << "\n";

    return 0;
}

```