

15

Null object Pattern

(LLD of handling NULL)

object → Value on Null

Problem:

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

Code will throw Null Pointer exception, if Vehicle object is NULL.

Minimum

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());  
    }  
}
```

What if 1000s of Methods & lakhs of Methods?

So, Kitne if Condition lagao ge?

So, How can we avoid (!= NULL)

Here comes the "NULL OBJECT DESIGN PATTERN"

- A null Object replaces NULL return type.
- No need to put IF CHECK for Checking NULL.
- Null Object reflects do Nothing or Default behavior.

if (c) {
 return Vehicle;
}
return NULL;
NULLobj;



