**Open/Closed Principle (OCP)**

A class, module, or function should be **open for extension** but **closed for modification**.

1. You should be able to **add new behaviour** (extend functionality) without **changing existing code**.
2. Instead of modifying existing classes, you **extend** them using **inheritance** or **interfaces**.

CODE EXAMPLE:

**VIOLATES**

public class DiscountCalculator {

public double calculateDiscount(String customerType, double amount) {

if (customerType.equals("Regular")) {

return amount \* 0.05;

} else if (customerType.equals("Premium")) {

return amount \* 0.1;

} else if (customerType.equals("VIP")) {

return amount \* 0.2;

}

return 0;

}

}

**Problem:**

* Every time a new customer type is added, you have to **modify** calculateDiscount.
* This breaks the **Open/Closed Principle**.

ADHERES:

//create an interface

public interface DiscountStrategy {

double getDiscount(double amount);

}

//create classes for each customer type

public class RegularCustomer implements DiscountStrategy {

public double getDiscount(double amount) {

return amount \* 0.05;

}

}

public class PremiumCustomer implements DiscountStrategy {

public double getDiscount(double amount) {

return amount \* 0.1;

}

}

public class VIPCustomer implements DiscountStrategy {

public double getDiscount(double amount) {

return amount \* 0.2;

}

}

//creating calculating class that uses interface

public class DiscountCalculator {

public double calculateDiscount(DiscountStrategy customer, double amount) {

return customer.getDiscount(amount);

}

}

//create main method

public class Main {

public static void main(String[] args) {

DiscountCalculator calculator = new DiscountCalculator();

DiscountStrategy regular = new RegularCustomer();

DiscountStrategy vip = new VIPCustomer();

System.out.println("Regular discount: " + calculator.calculateDiscount(regular, 1000)); // 50.0

System.out.println("VIP discount: " + calculator.calculateDiscount(vip, 1000)); // 200.0

}

}

To add a new customer type, one just needs to write a new class. Theres no need to touch DiscountCalculator hence it being easy to extend.