**Assignment 1: Implement the Strategy Pattern in a Payment System**

### **Objective**

The goal of this assignment is to practice the **Strategy Design Pattern** in Java. You will create a payment processing system that supports multiple payment methods (e.g., Credit Card, PayPal, UPI) and allows switching between them dynamically at runtime.

### **Problem Statement**

A company’s e-commerce application needs a flexible payment system. Depending on the user’s choice, the system should be able to process payments through different methods.

The current requirements are:

1. Support for **Credit Card**, **PayPal**, and **UPI** payments.
2. The ability to **switch payment methods dynamically at runtime**.
3. A clear separation between the **payment context** (e.g., PaymentProcessor) and the **payment strategies** (e.g., PayByCreditCard, PayByPaypal, etc.).

### **Tasks**

1. **Define the Strategy Interface**
   1. Create an interface PaymentStrategy with a method:

***void pay(double amount);***

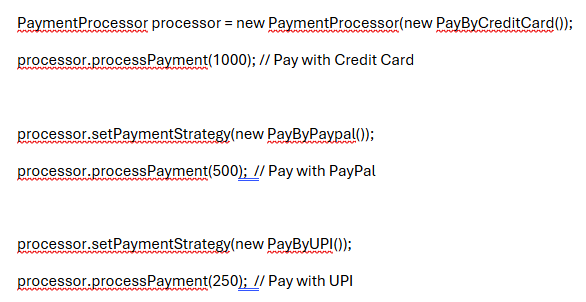
1. **Implement Concrete Strategies**
   * PayByCreditCard
   * PayByPaypal
   * PayByUPI  
      Each should implement the pay method and print out how the payment is being processed.
2. **Create the Context Class**
   * Class: PaymentProcessor
   * Has a private PaymentStrategy field.
   * Provides methods:

***public void setPaymentStrategy(PaymentStrategy strategy)***

***public void processPayment(double amount)***

1. **Demonstrate Usage**

* In a main method (or a test class), do the following:



### **Deliverables**

* PaymentStrategy.java (Interface)
* PayByCreditCard.java, PayByPaypal.java, PayByUPI.java (Concrete strategies)
* PaymentProcessor.java (Context class)
* Main.java (or TestPayment.java) to demonstrate the functionality

**Expected Output**

***Paid 1000.0 using Credit Card***

***Paid 500.0 using PayPal***

***Paid 250.0 using UPI***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****