**Exercise 8: Implementing the Strategy Pattern**

**Scenario:**

You are developing a payment system where different payment methods (e.g., Credit Card, PayPal) can be selected at runtime. Use the Strategy Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **StrategyPatternExample**.
2. **Define Strategy Interface:**
   * Create an interface PaymentStrategy with a method **pay()**.
3. **Implement Concrete Strategies:**
   * Create classes **CreditCardPayment**, **PayPalPayment** that implement **PaymentStrategy**.
4. **Implement Context Class:**
   * Create a class **PaymentContext** that holds a reference to **PaymentStrategy** and a method to execute the strategy.
5. **Test the Strategy Implementation:**
   * Create a test class to demonstrate selecting and using different payment strategies.

**PaymentStrategy Interface**: The PaymentStrategy interface defines the pay method that all payment strategies must implement.

**CreditCardPayment Class**: This class implements the PaymentStrategy interface and provides the implementation for processing credit card payments.

**PayPalPayment Class**: This class implements the PaymentStrategy interface and provides the implementation for processing PayPal payments.

**PaymentContext Class**: This class holds a reference to PaymentStrategy and has a setPaymentStrategy method to set the payment strategy at runtime. The pay method executes the current payment strategy.

**Testing the Strategy Pattern**: In the StrategyPatternTest class, we create instances of PaymentContext and payment strategy classes, set different payment strategies at runtime, and execute the pay method to demonstrate the use of different payment strategies.