**Module 1 - Design Patterns and Principles**

**Exercise 4: Implementing the Adapter Pattern**

**Aim:**

To develop a payment processing system that needs to integrate with multiple third-party payment gateways with different interfaces using the Adapter Pattern to achieve this.

**Code:**

interface PaymentProcessor {

    void processPayment(int amount);

}

class Razorpay {

    public void payUsingRazor(int amount) {

        System.out.println("Processing Rs." + amount + " using Razorpay");

    }

}

class Paytm {

    public void payUsingPaytm(int amount) {

        System.out.println("Processing Rs." + amount + " using Paytm");

    }

}

class RazorpayAdapter implements PaymentProcessor {

    private Razorpay razorpay = new Razorpay();

    public void processPayment(int amount) {

        razorpay.payUsingRazor(amount);

    }

}

class PaytmAdapter implements PaymentProcessor {

    private Paytm paytm = new Paytm();

    public void processPayment(int amount) {

        paytm.payUsingPaytm(amount);

    }

}

public class AdapterPatternExample {

    public static void main(String[] args) {

        PaymentProcessor razor = new RazorpayAdapter();

        razor.processPayment(1000);

        PaymentProcessor paytm = new PaytmAdapter();

        paytm.processPayment(1000);

    }

}

**Output:**

