**Exercise 4: Implementing the Adapter Pattern**

**PaymentProcessor.java**

public interface PaymentProcessor {

    void processPayment(double amount);

    String getPaymentGateway();

}

**Gpay.java**

public class GPay {

    public void makePayment(double amount) {

        System.out.println("Processing payment of " + amount + " through GPay.");

    }

}

**Paypal.java**

public class PayPal {

    public void makePayment(double amount) {

        System.out.println("Processing payment of " + amount + " through PayPal.");

    }

}

**GPayAdapter.java**

public class GpayAdapter implements PaymentProcessor {

    private final GPay gPay;

    public GpayAdapter(GPay gPay) {

        this.gPay = gPay;

    }

    @Override

    public void processPayment(double amount) {

        gPay.makePayment(amount);

    }

    @Override

    public String getPaymentGateway() {

        return "GPay";

    }

}

**PayPalAdapter.java**

public class PayPalAdapter implements PaymentProcessor {

    private final PayPal payPal;

    public PayPalAdapter(PayPal payPal) {

        this.payPal = payPal;

    }

    public void processPayment(double amount) {

        payPal.makePayment(amount);

    }

    public String getPaymentGateway() {

        return "PayPal";

    }

}

**Test.java**

public class Test {

    public static void main(String[] args) {

        PaymentProcessor paymentProcessor = new GpayAdapter(new GPay());

        paymentProcessor.processPayment(100.0);

        System.out.println("Using payment gateway: " + paymentProcessor.getPaymentGateway());

        PaymentProcessor paymentProcessor2 = new PayPalAdapter(new PayPal());

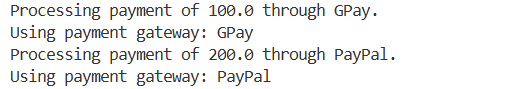
        paymentProcessor2.processPayment(200.0);

        System.out.println("Using payment gateway: " + paymentProcessor2.getPaymentGateway());

    }

}

**Output**

****