**Scenario 1: Multi-Tiered Payment System**

**Problem Statement**

Design a **multi-tiered payment system** for an e-commerce platform that supports:

1. Multiple payment methods like **Credit Card**, **PayPal**, and **UPI**.
2. Different **payment processing tiers** such as **Regular**, **Premium**, and **Enterprise**.

Each payment tier has different processing rules:

* **Regular Tier**: Processes payments with standard fees.
* **Premium Tier**: Processes payments with reduced fees.
* **Enterprise Tier**: Processes payments with no fees.

The system must:

* Use interfaces to define common behavior for payment methods and tiers.
* Allow a flexible combination of payment methods and tiers using polymorphism.
* Output the final payment details (amount, fees, etc.).

**Assignment Requirements**

1. Define an interface PaymentMethod with methods:
   * double calculatePayment(double amount).
   * void displayDetails().
2. Define an interface PaymentTier with a method:
   * double applyFees(double amount).
3. Implement multiple payment methods:
   * CreditCardPayment.
   * PayPalPayment.
   * UPIPayment.
4. Implement multiple payment tiers:
   * RegularTier.
   * PremiumTier.
   * EnterpriseTier.
5. In the main class:
   * Allow the user to choose a payment method and tier.
   * Calculate and display the final payment amount.

**Sample Input**

Payment Method: PayPal

Payment Tier: Premium

Transaction Amount: 1000.0

**Sample Output**

Payment Method: PayPal

Payment Tier: Premium

Original Amount: 1000.0

Fees Applied: 10.0

Final Amount: 1010.0

**Scenario 2: Modular Notification System for Distributed Applications**

**Problem Statement**

Build a **modular notification system** for a distributed application where:

1. Notifications can be sent via **Email**, **SMS**, and **Push Notifications**.
2. Each notification type supports multiple **delivery strategies**, such as:
   * **Instant Delivery**: Sends the notification immediately.
   * **Scheduled Delivery**: Schedules the notification for a specific time.
   * **Batch Delivery**: Sends multiple notifications at once.

The system must:

* Use interfaces to define common behavior for notification types and delivery strategies.
* Allow flexibility to combine notification types with delivery strategies dynamically.
* Ensure the system can scale to support new notification types or strategies.

**Assignment Requirements**

1. Define an interface Notification with methods:
   * void send(String recipient, String message).
   * void displayInfo().
2. Define an interface DeliveryStrategy with a method:
   * void deliver(String recipient, String message).
3. Implement multiple notification types:
   * EmailNotification.
   * SMSNotification.
   * PushNotification.
4. Implement multiple delivery strategies:
   * InstantDelivery.
   * ScheduledDelivery.
   * BatchDelivery.
5. In the main class:
   * Combine notification types with delivery strategies dynamically.
   * Send notifications based on the chosen combination.

**Sample Input**

Notification Type: Email

Delivery Strategy: Scheduled

Recipient: user@example.com

Message: Your order has been shipped!

**Sample Output**

Notification Type: Email

Delivery Strategy: Scheduled

Recipient: user@example.com

Message: Your order has been scheduled for delivery at 3:00 PM.