**Exercise 7: Implementing the Observer Pattern**

**Scenario:**

You are developing a stock market monitoring application where multiple clients need to be notified whenever stock prices change. Use the Observer Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **ObserverPatternExample**.
2. **Define Subject Interface:**
   * Create an interface **Stock** with methods to **register**, **deregister**, and **notify** observers.
3. **Implement Concrete Subject:**
   * Create a class **StockMarket** that implements **Stock** and maintains a list of observers.
4. **Define Observer Interface:**
   * Create an interface Observer with a method **update().**
5. **Implement Concrete Observers:**
   * Create classes **MobileApp**, **WebApp** that implement Observer.
6. **Test the Observer Implementation:**
   * Create a test class to demonstrate the registration and notification of observers.

**CODE:**

using System;

interface IStock

{

void RegisterObserver(IObserver observer);

void DeregisterObserver(IObserver observer);

void NotifyObservers(string stockName, decimal price);

}

class StockMarket : IStock

{

private List<IObserver> observers = new List<IObserver>();

public void RegisterObserver(IObserver observer)

{

observers.Add(observer);

}

public void DeregisterObserver(IObserver observer)

{

observers.Remove(observer);

}

public void NotifyObservers(string stockName, decimal price)

{

foreach (var observer in observers)

{

observer.Update(stockName, price);

}

}

public void UpdateStockPrice(string stockName, decimal price)

{

Console.WriteLine($"Stock price updated: {stockName} - ${price}");

NotifyObservers(stockName, price);

}

}

interface IObserver

{

void Update(string stockName, decimal price);

}

class MobileApp : IObserver

{

public void Update(string stockName, decimal price)

{

Console.WriteLine($"[MobileApp] {stockName} updated to ${price}");

}

}

class WebApp : IObserver

{

public void Update(string stockName, decimal price)

{

Console.WriteLine($"[WebApp] {stockName} updated to ${price}");

}

}

class Program

{

static void Main()

{

StockMarket stockMarket = new StockMarket();

IObserver mobileApp = new MobileApp();

IObserver webApp = new WebApp();

stockMarket.RegisterObserver(mobileApp);

stockMarket.RegisterObserver(webApp);

stockMarket.UpdateStockPrice("AAPL", 182.50m);

stockMarket.UpdateStockPrice("GOOG", 2750.30m);

// Deregister one observer and update again

stockMarket.DeregisterObserver(webApp);

stockMarket.UpdateStockPrice("MSFT", 305.00m);

}

}

**OUTPUT:**

