**Exercise 6: Implementing the Proxy Pattern**

**Scenario:**

You are developing an image viewer application that loads images from a remote server. Use the Proxy Pattern to add lazy initialization and caching.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **ProxyPatternExample**.
2. **Define Subject Interface:**
   * Create an interface Image with a method **display()**.
3. **Implement Real Subject Class:**
   * Create a class **RealImage** that implements Image and loads an image from a remote server.
4. **Implement Proxy Class:**
   * Create a class **ProxyImage** that implements Image and holds a reference to RealImage.
   * Implement lazy initialization and caching in **ProxyImage**.
5. **Test the Proxy Implementation:**
   * Create a test class to demonstrate the use of **ProxyImage** to load and display images.

Code –

package Design\_Patterns\_And\_Principles.ProxyPatternExample;

interface Image {

    void display();

}

 class RealImage implements Image {

    private String imagePath;

    public RealImage(String imagePath) {

        this.imagePath = imagePath;

        loadImageFromServer();

    }

    private void loadImageFromServer() {

        // Simulate loading image from a remote server

        System.out.println("Loading image from server: " + imagePath);

    }

    @Override

    public void display() {

        System.out.println("Displaying image: " + imagePath);

    }

}

 class ProxyImage implements Image {

    private RealImage realImage;

    private String imagePath;

    public ProxyImage(String imagePath) {

        this.imagePath = imagePath;

    }

    @Override

    public void display() {

        if (realImage == null) {

            realImage = new RealImage(imagePath);

        }

        realImage.display();

    }

}

public class ProxyPatternDemo {

    public static void main(String[] args) {

        Image image1 = new ProxyImage("image1.jpg");

        Image image2 = new ProxyImage("image2.jpg");

        // Image is loaded and displayed from the remote server

        image1.display();

        System.out.println();

        // Image is already loaded, so it's displayed directly from cache

        image1.display();

        System.out.println();

        // Another image is loaded and displayed from the remote server

        image2.display();

    }

}