**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.

**Step 1:** Create a New Java Project

**Step 2:** Define Subject Interface

// Image

interface Image {

void display();

}

**Step 3**: Implement Real Subject Class

//RealImage

class RealImage implements Image {

private String imageLink;

public RealImage(String imageLink) {

this.imageLink = imageLink;

loadImage(imageLink);

}

private void loadImage(String imageLink) {

System.out.println("The image is being loaded from the site " + imageLink);

}

@Override

public void display() {

System.out.println("Displaying " + imageLink);

}

}

**Step 4:** Implement Proxy Class

// ProxyImage

class ProxyImage implements Image {

private String imageLink;

private RealImage realImage;

public ProxyImage(String imageLink) {

this.imageLink = imageLink;

}

@Override

public void display() {

if (realImage == null) {

realImage = new RealImage(imageLink);

}

realImage.display();

System.out.println();

}

}

**Step 5:** Test the Proxy Implementation

public class ProxyPatternExample {

public static void main(String[] args) {

Image image1 = new ProxyImage("http://example.com/image1.jpg");

Image image2 = new ProxyImage("http://example.com/image2.jpg");

// Display the images

image1.display(); // Loads from server and displays

image2.display(); // Loads from server and displays

image1.display(); // Displays from cache

image2.display(); // Displays from cache

}

}

**SAMPLE OUTPUT:**

