**Module 1 - Design Patterns and Principles**

**Exercise 6: Implementing the Proxy Pattern**

**Aim:**

To develop an image viewer application that loads images from a remote server using the Proxy Pattern to add lazy initialization and caching.

**Code:**

interface Image {

    void display();

}

class RealImage implements Image {

    private String filename;

    public RealImage(String filename) {

        this.filename = filename;

        loadFromDisk();

    }

    private void loadFromDisk() {

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

    }

    public void display() {

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

    }

}

class ProxyImage implements Image {

    private String filename;

    private RealImage realImage;

    public ProxyImage(String filename) {

        this.filename = filename;

    }

    public void display() {

        if (realImage == null) {

            realImage = new RealImage(filename);

        }

        realImage.display();

    }

}

public class TestProxy {

    public static void main(String[] args) {

        Image img = new ProxyImage("pic1.jpg");

        img.display();

        img.display();

    }

}

**Output:**

****