**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 6: Implementing the Proxy Pattern**

**Step 1: Subject Interface – Image.java**

public interface Image {

void display();

}

**Step 2: Real Subject Class – RealImage.java**

public class RealImage implements Image {

private String filename;

public RealImage(String filename) {

this.filename = filename;

loadFromServer();

}

private void loadFromServer() {

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

}

public void display() {

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

}

}

**Step 3: Proxy Class – ProxyImage.java**

public class ProxyImage implements Image {

private RealImage realImage;

private String filename;

public ProxyImage(String filename) {

this.filename = filename;

}

public void display() {

if (realImage == null) {

realImage = new RealImage(filename); // lazy loading

}

realImage.display(); // cached call

}

}

**Step 4: Test Class – TestProxyPattern.java**

public class TestProxyPattern {

public static void main(String[] args) {

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

// Image will be loaded now (first time)

image1.display();

System.out.println();

// Image won't be loaded again (cached)

image1.display();

}

}

**OUTPUT:**

