### ****Exercise 6: Implementing the Proxy Pattern****

#### ****Scenario****

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

### ****Step-by-step Implementation****

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

Create a new Java project named ProxyPatternExample.

#### ****Step 2: Define Subject Interface****

public interface Image {

void display();

}

#### ****Step 3: Implement Real Subject Class****

public class RealImage implements Image {

private String filename;

public RealImage(String filename) {

this.filename = filename;

loadFromDisk();

}

private void loadFromDisk() {

System.out.println("Loading " + filename);

}

@Override

public void display() {

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

}

}

#### ****Step 4: Implement Proxy Class****

public class ProxyImage implements Image {

private RealImage realImage;

private String filename;

public ProxyImage(String filename) {

this.filename = filename;

}

@Override

public void display() {

if (realImage == null) {

realImage = new RealImage(filename);

}

realImage.display();

}

}

#### ****Step 5: Test the Proxy Implementation****

public class TestProxyPattern {

public static void main(String[] args) {

Image image = new ProxyImage("photo1.jpg");

// Image will be loaded from disk

image.display();

System.out.println();

// Image will not be loaded again

image.display();

}

}

OUTPUT:

