**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:**

interface IImage

{

void Display();

}

class RealImage : IImage

{

private string filename;

public RealImage(string filename)

{

this.filename = filename;

LoadFromDisk();

}

private void LoadFromDisk() => Console.WriteLine($"Loading {filename}");

public void Display() => Console.WriteLine($"Displaying {filename}");

}

class ProxyImage : IImage

{

private RealImage realImage;

private string filename;

public ProxyImage(string filename) => this.filename = filename;

public void Display()

{

if (realImage == null)

realImage = new RealImage(filename);

realImage.Display();

}

}

class Program

{

static void Main()

{

IImage image = new ProxyImage("photo.jpg");

image.Display();

image.Display();

}

}

**OUTPUT:**

