## Overview

In this practice, you will take an existing application and refactor the code to implement the Singleton design pattern.

## Summary

You are working on server software that synchronizes with other servers. Your task is to create a Singleton class which stores the hostnames of the servers to connect with. The server list is declared in a static initialization block.

## Tasks

1. Open the `Singleton04-03Prac` project.
   a. Select File > Open Project.
   b. Browse to `\home\oracle\labs\04-Polymorphism\practices\practice3`.
   c. Select `Singleton04-03Prac` and click Open Project.
2. Expand the project directories.
3. Modify the `PeerSingleton` class to implement the Singleton design pattern.
   a. Open the `PeerSingleton.java` file (under the `com.example` package).
   b. Change the constructor's access level to `private`.
   c. Add a new field named `instance`. The field should be:
      i. `private`
      ii. Marked `static`
      iii. Marked `final`
      iv. Type of `PeerSingleton`
      v. Initialized to a new `PeerSingleton` instance
   d. Create a static method named `getInstance` that returns the value stored in the `instance` field.
4. Modify the `Main` class to use the singleton.
   a. Open the `Main.java` file (under the `com.example` package).
   b. Perform the following steps in the `main` method:
      1) Create a `PeerSingleton` reference named `peerList01` and initialize it using the `getInstance` method.
      2) Create a second `PeerSingleton` reference named `peerList02` and initialize it using the `getInstance` method.
      3) Display the host names by invoking `getHostNames` on `peerList01` in a `for` loop.
      4) Next, display the host names by invoking `getHostNames` on `peerList02` in a `for` loop.
5. Run the project. You should see a list of host names.