In this project, you are required to write client and server programs to demonstrate Java Remote Method Invocation (RMI). Java RMI is a mechanism that allows one to invoke a method on an object that exists in another address space. The other address space could be on the same machine or a different one. The RMI mechanism is basically an object-oriented RPC mechanism.

There are three processes that participate in supporting remote method invocation.

1. The *Client* is the process that is invoking a method on a remote object.
2. The *Server* is the process that owns the remote object. The remote object is an ordinary object in the address space of the server process.
3. The *Object Registry* is a name server that relates objects with names. Objects are *registered* with the Object Registry. Once an object has been registered, one can use the Object Registry to obtain access to a remote object using the name of the object.

There are two kinds of classes that are used in Java RMI. You should demonstrate the use of both in your program:

1. A *Remote* class is one whose instances can be used remotely. An object of such a class can be referenced in two different ways:
   1. Within the address space where the object was constructed, the object is an ordinary object which can be used like any other object.
   2. Within other address spaces, the object can be referenced using an *object handle*. While there are limitations on how one can use an object handle compared to an object, for the most part one can use object handles in the same way as an ordinary object.

For simplicity, an instance of a Remote class will be called a *remote object*.

1. A *Serializable* class is one whose instances can be copied from one address space to another. An instance of a Serializable class will be called a *serializable object*. In other words, a serializable object is one that can be marshaled. Note that this concept has no connection to the concept of serializability in database management systems.

If a serializable object is passed as a parameter (or return value) of a remote method invocation, then the value of the object will be copied from one address space to the other. By contrast if a remote object is passed as a parameter (or return value), then the object handle will be copied from one address space to the other.

Description of project:

This project is done in a group of only 2 persons per group. One group member writes the client program, the other member of the group writes the server program. Both members need to work together to make sure that client and server programs can run on separate machines.

You need to write a Java Client program that queries for the last name of students in this class providing their first name. The list of first names and last names are kept only at the server location. If there is no person with a requested first name, it should simply report that there is no one with such name in class. If there are more than one person in class with the same first name, then it will retrieve all the students with that first name. Your program should run successfully to demonstrate the RMI on two separate machines.

For more details on Java RMI refer to:

<http://www.oracle.com/technetwork/java/index.html>