Experiment: 2

Aim: Write a program to implement Remote Method Invocation.

Theory:

Remote Method Invocation (RMI) is an application programming interface (API) in the Java programming language and development environment. It allows objects on one computer or Java Virtual Machine (JVM) to interact with objects running on a different JVM in a distributed network. Another way to say this is that RMI provides a way to create distributed Java applications through simple method calls.

RMI is the Java version of what's known as a remote procedure call (RPC), but with the additional ability to pass one or more objects along with the request. It enables remote communication between applications using two objects -- stub and skeleton -- and is supplied as part of Sun Microsystems' Java Development Kit (JDK) in the package java.rmi.

Code:

1) Interface:

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Hello extends Remote {
        void printMsg() throws RemoteException;
}
```

2) Implementation of Remote Interface:

```
public class ImplExample implements Hello {
        public void printMsg() {
                System.out.println("This is an example RMI program");
        }
}
```

3) Server Code:

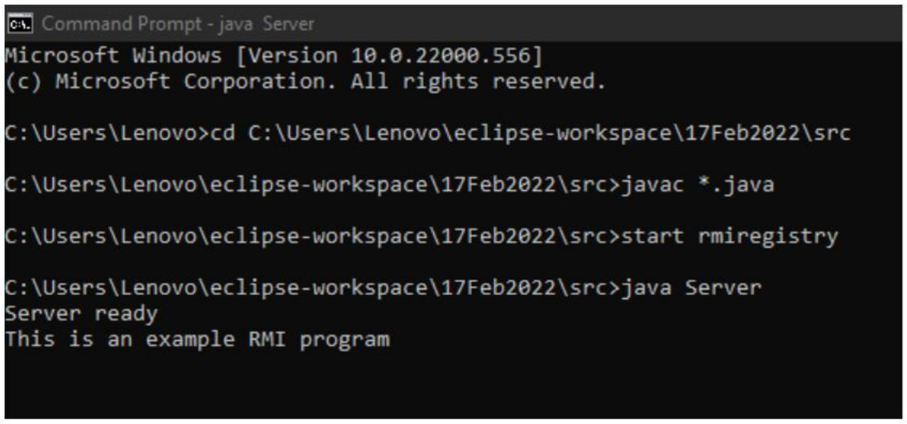
```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;

public class Server extends ImplExample {
		public Server() {
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
			// Instantiating the implementation class
			ImplExample obj = new ImplExample();

			// Exporting the object of implementation class
			// (here we are exporting the remote object to the stub)
			Hello stub = (Hello) UnicastRemoteObject.exportObject(obj, 0);

			// Binding the remote object (stub) in the registry
			Registry registry = LocateRegistry.getRegistry();

			registry.bind("Hello", stub);
			System.err.println("Server ready");
		}
		catch (Exception e) {
			System.err.println("Server exception: " + e.toString());
			e.printStackTrace();
		}
	}
}
```

4) Client Code:

```
	import java.rmi.registry.LocateRegistry;
	import java.rmi.registry.Registry;

	public class Client {
		private Client() {
		}

		public static void main(String[] args) {
			// TODO Auto-generated method stub
			try {

			// Getting the registry
			Registry registry = LocateRegistry.getRegistry(null);

			// Looking up the registry for the remote object
			Hello stub = (Hello) registry.lookup("Hello");

			// Calling the remote method using the obtained object
```

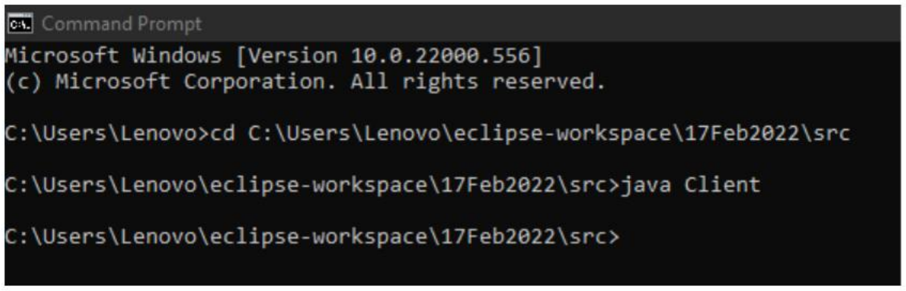
```
            stub.printMsg();

            // System.out.println("Remote method invoked");
            } catch (Exception e) {
                System.err.println("Client exception: " + e.toString());
                e.printStackTrace();
            }
        }
```

Output:



Command Prompt - java Server

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>cd C:\Users\Lenovo\eclipse-workspace\17Feb2022\src

C:\Users\Lenovo\eclipse-workspace\17Feb2022\src>javac *.java

C:\Users\Lenovo\eclipse-workspace\17Feb2022\src>start rmiregistry

C:\Users\Lenovo\eclipse-workspace\17Feb2022\src>java Server
Server ready
This is an example RMI program
```



Command Prompt

```
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>cd C:\Users\Lenovo\eclipse-workspace\17Feb2022\src

C:\Users\Lenovo\eclipse-workspace\17Feb2022\src>java Client

C:\Users\Lenovo\eclipse-workspace\17Feb2022\src>
```