Project 2

COSC 3657

Agam Jammu

The code represents a basic implementation of a Remote Method Invocation (RMI) system, which allows a client to remotely invoke methods on a server. It is a good example of how RMI works and how the RMI components interact with each other.

The code makes use of the following libraries:

1. java.rmi: This library is used to implement RMI functionality in the code. It provides the classes and interfaces required to create and manage RMI objects, including remote interfaces, stubs, and skeletons.
2. java.rmi.server: This library is used to provide the implementation of RMI servers. It provides the UnicastRemoteObject class, which is used by the Server class to create a remote object.

The code consists of three separate files, each of which has a specific purpose:

Server.java: This file contains the Server class, which is responsible for providing the remote methods that the client can invoke. The Server class extends the UnicastRemoteObject class, which provides the implementation for RMI servers. The Server class implements the ServerInterface interface, which defines the two methods that the server provides: getFunctionList() and calculate(). getFunctionList() returns an array of strings that contains the available arithmetic operations, while calculate() takes three parameters: an arithmetic operation, operand1, and operand2, and performs the operation on the two operands. The result of the operation is returned to the client.

Client.java: This file contains the Client class, which represents the client side of the RMI system. The Client class prompts the user to select an operation from the list of available functions and prompts the user for the two operands required to perform the operation. The Client class then calls the calculate() method on the remote server object and displays the result to the user.

ServerInterface.java: This file contains the ServerInterface interface, which defines the two methods that the Server class implements. This interface is used by both the Server and Client classes to define the available methods for remote invocation.

To run the code, follow these instructions:

1. Ensure that Java is installed on your machine.
2. Download the three Java files (Server.java, Client.java, ServerInterface.java) and save them in a directory on your machine.
3. Compile the Server.java file by typing the following command in the terminal/command prompt: javac Server.java
4. Compile the Client.java file by typing the following command in the terminal/command prompt: javac Client.java
5. Start the RMI registry by typing the following command in the terminal/command prompt: rmiregistry
6. In a new terminal/command prompt window, start the server by typing the following command: java Server
7. In another terminal/command prompt window, start the client by typing the following command: java Client
8. The client will prompt you to select an arithmetic operation from the list of available functions. Enter the number corresponding to the operation you want to use.
9. The client will then prompt you to enter two operands. Enter the operands as integers and press enter.
10. The client will display the result of the calculation.
11. You can repeat steps 8-10 to perform additional calculations.
12. To stop the RMI registry, press CTRL+C in the terminal/command prompt window where the registry is running.

Running Screenshots:

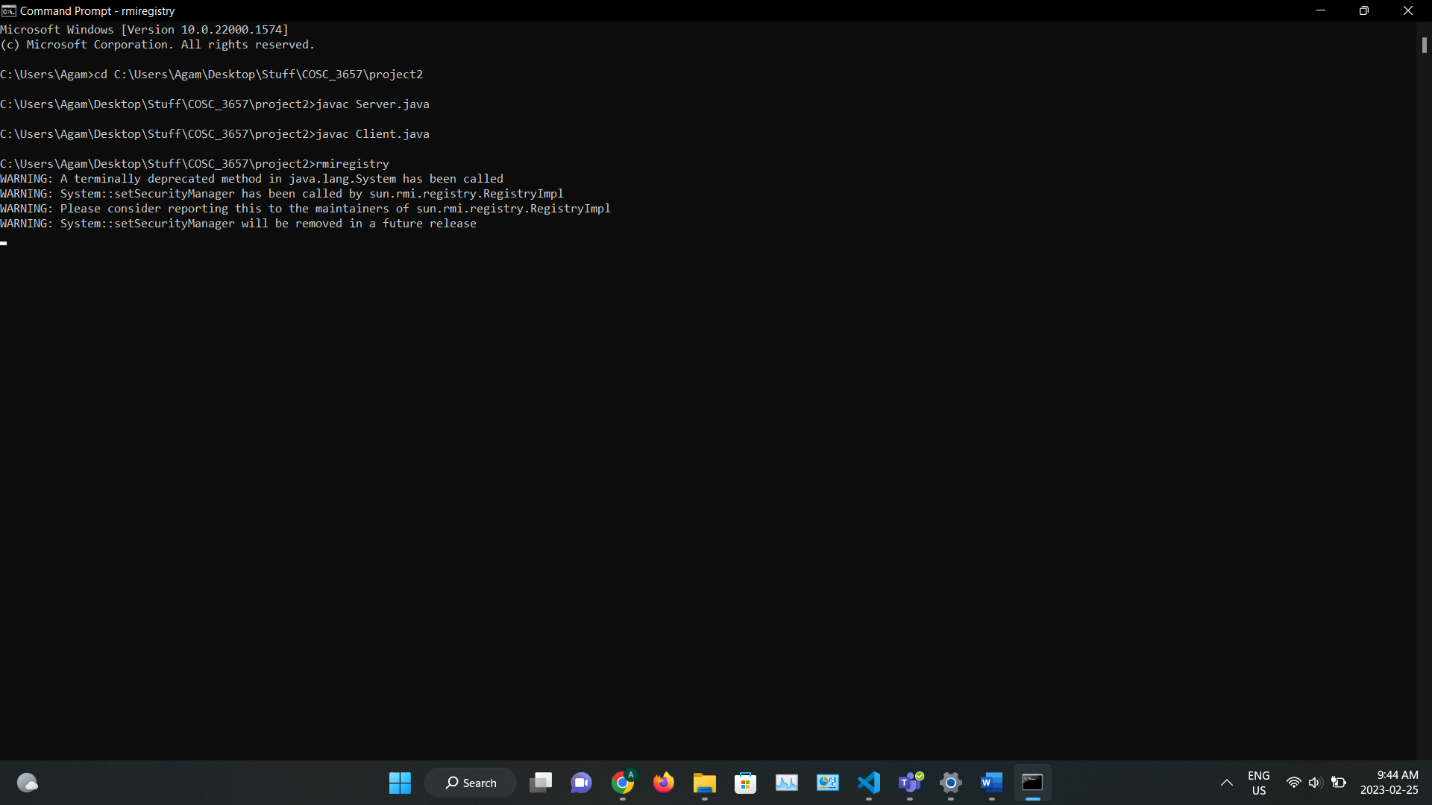


Figure 1

In Figure 1, Server,java and Client.java have been compiled and rmiregistry has been started,

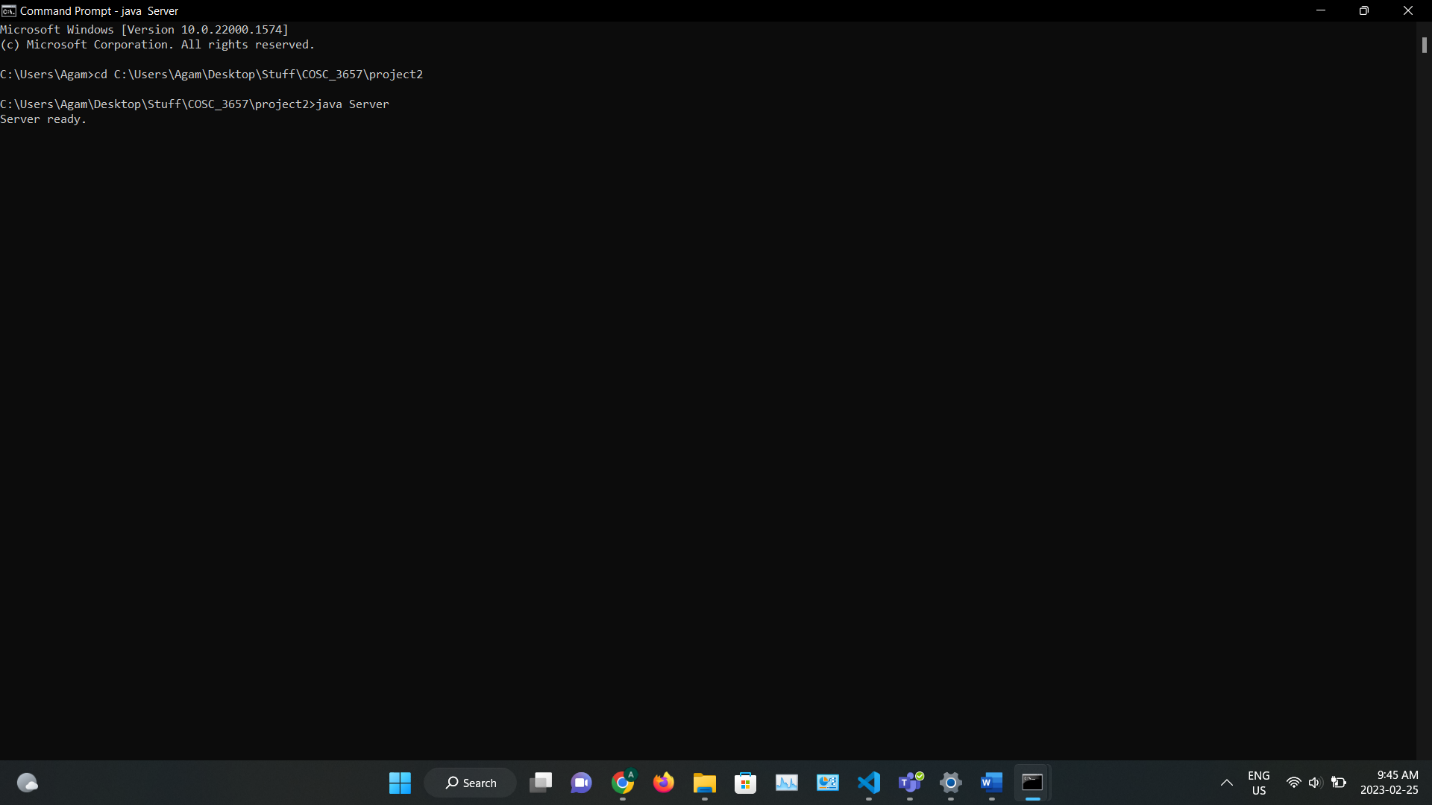


Figure 2

In figure 2, the server has been started.

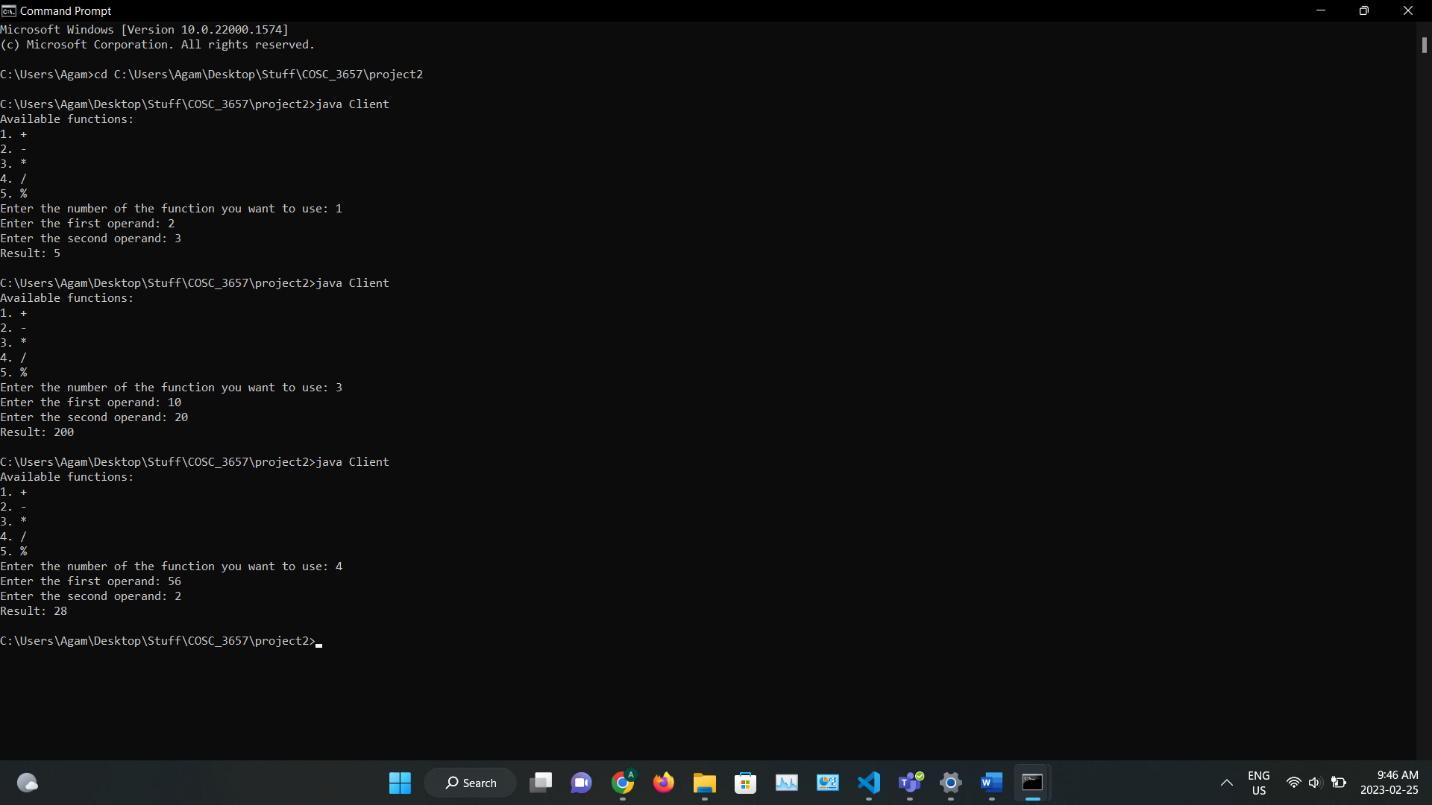


Figure 3

In Figure 3, the client has been started.