Fr. Conceicao Rodrigues College of Engineering

Department of Computer Engineering

Academic Term: July-Nov 2023-24

Class: T.E. (Computer B)
Subject Name: Computer Network Lab
Subject Code: CSL 502

Experiment No:	5
Date of Performance:	17/08/2023
Roll No:	9614
Name of the Student:	Aqib Firdous Khan

AIM: Java program for Socket Programming

THEORY:
Java Socket Programming

- Java Socket programming is used for communication between the applications running on different JRE.
- Java Socket programming can be connection-oriented or connection-less.
- Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.

The client in socket programming must know two pieces of information:

a. IP Address of Server, and
b. Port number.

Here, we are going to make one-way client and server communication. In this application, the client sends a message to the server, and the server reads the message and prints it. Here, two classes are being used: Socket and ServerSocket.

The Socket class is used to communicate client and server. Through this class, we can read and write messages. The ServerSocket class is used on the server side. The accept() method of ServerSocket class blocks the console until the client is connected. After the successful connection of the client, it returns the instance of Socket on the server side.

#Socket class

A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.

#ServerSocket class

The ServerSocket class can be used to create a server socket. This object is used to establish communication with the clients.

Creating Server:
To create the server application, we need to create the instance of ServerSocket class. Here, we are using 6666 port number for the communication between the client and server. You may also choose any other port number. The accept() method waits for the client. If clients connects with the given port number, it returns an instance of Socket.

```
ServerSocket ss=new ServerSocket(6666);
Socket s=ss.accept();//establishes a connection and waits for the client
```

Creating Client:
To create the client application, we need to create the instance of Socket class. Here, we need to pass the IP address or hostname of the Server and a port number. Here, we are using "localhost" because our server is running on the same system.

```
Socket s=new Socket("localhost",6666);
```

Code:

MyServer.java file

```
import java.io.*;
import java.net.*;
public class MyServer{
        public static void main(String[] args){
        try
        {
                ServerSocket ss=new ServerSocket(6666);
                Socket s=ss.accept();//establishes connection
                DataInputStream dis=new DataInputStream(s.getInputStream());
                String str=(String)dis.readUTF();
                System.out.println("message= "+str);
                ss.close();
                }
                catch(Exception e){System.out.println(e);}
        }
    }
```

MyClient.java file

```
import java.io.*;
import java.net.*;
public class MyClient
{
    public static void main(String[] args)
    {
        try
        {
        Socket s=new Socket("localhost",6666);
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        dout.writeUTF("Hello Server");
        dout.flush();
        dout.close();
        s.close();
        }catch(Exception e){System.out.println(e);}
    }
}
```

Output:

To execute this program open two command prompts and execute each program at each command prompt as displayed in the below figures.

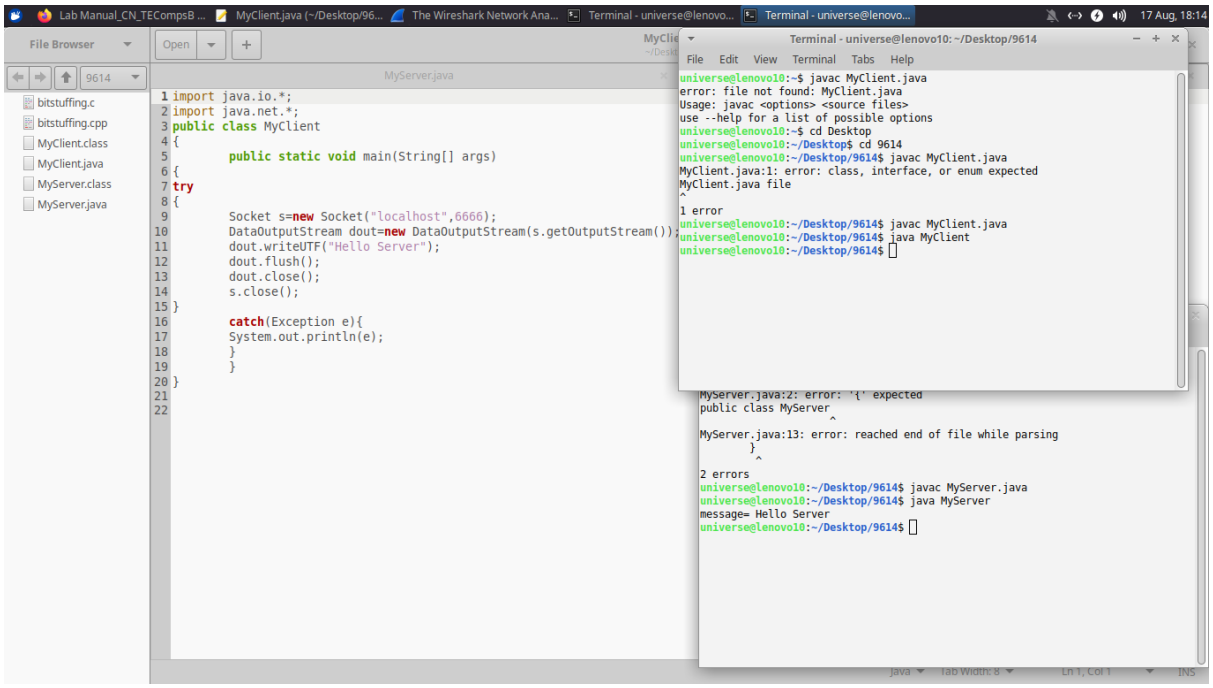
First, run the Myserver.java file in terminal/cmd,

Running MyServer.java

Then in new terminal/cmd run the MyClient.java file,

Running MyClient.java

As soon as you run the MyClient program a message is sent to the server and displayed in MyServer Terminal/CMD as shown below,



Message displayed in MyServer after running MyClient

CONCLUSION: So, in this experiment, we have successfully understood the concept of Socket Programming and implemented it using Java Programming.