CL-IX Assignment 1A

Roll No. 43104
Ajinkya Tikhe

Assignment 1A - TCP Java Sockets

Aim: To develop any distributed application through implementing client-server communication programs based on TCP Java Sockets.

Problem Statment: String Manipulation: Design a distributed application that consists of client-server communication using TCP, UDP & RMI techniques in Java. Multiple clients can simultaneously connect to the server and multiple clients submit two strings to the server and the server returns the concatenation of the given strings.

File 1:

Server.java

```
// A Java program for a Server
import java.net.*;
import java.io.*;

public class Server
{
	//initialize socket and input stream
	private Socket		socket = null;
	private ServerSocket server = null;
	private DataInputStream in	= null;

	// constructor with port
	public Server(int port)
	{
		// starts server and waits for a connection
		try
		{
```

```
        server = new ServerSocket(port);
        System.out.println("Server started");

        System.out.println("Waiting for a client ...");

        socket = server.accept();
        System.out.println("Client accepted");

        // takes input from the client socket
        in = new DataInputStream(
                new BufferedInputStream(socket.getInputStream()));

        String line = "";
        String line1 = "";

        for(int n=0; n<2; n++)
                {
                        try
                        {
                                line1 = in.readUTF();
                                line = line+line1;

                        }
                        catch(IOException i)
                        {
                                System.out.println(i);
                        }
                }

        System.out.println("Concatenation of the given strings : " + line);
        System.out.println("Closing connection");

        // close connection
        socket.close();
        in.close();
}
catch(IOException i)
{
        System.out.println(i);
```

```
		}
	}

	public static void main(String args[])
	{
		Server server = new Server(5000);
	}
}
```

—----------

File 2:

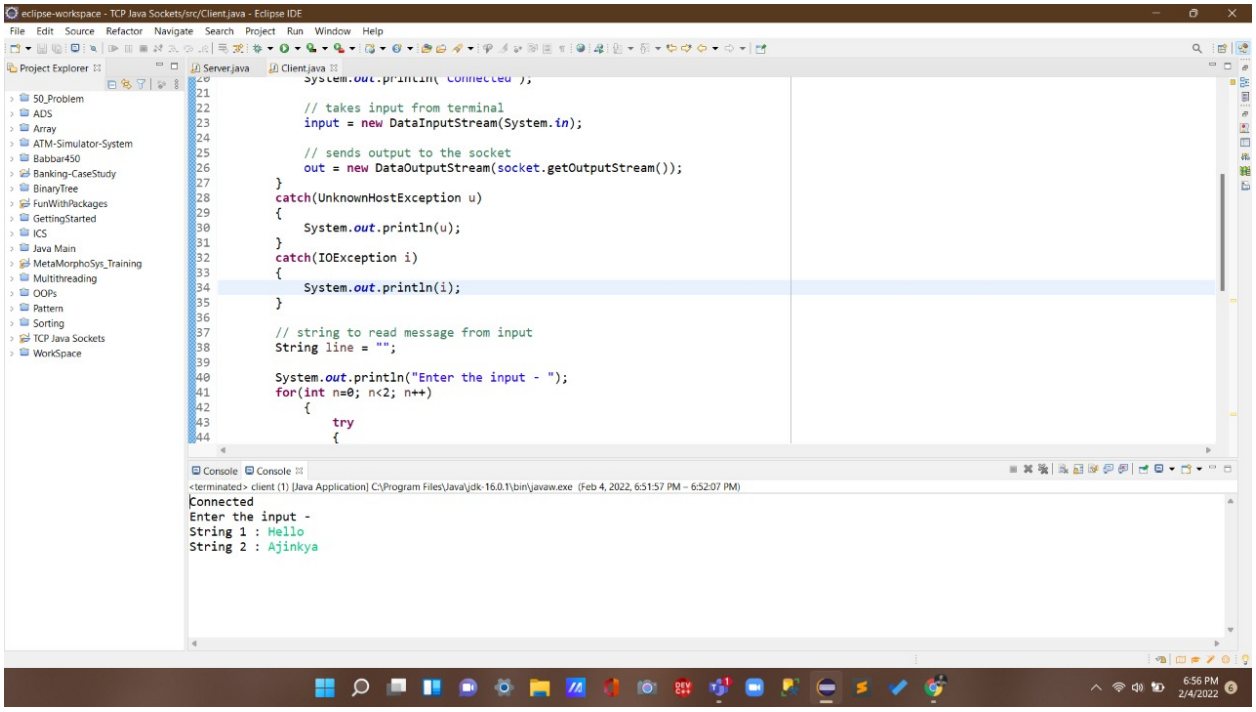
Client.java

```
// A Java program for a Client
import java.net.*;

import java.io.*;

public class Client
{
	// initialize socket and input output streams
	private Socket socket		= null;
	private DataInputStream input = null;
	private DataOutputStream out	= null;

	// constructor to put ip address and port
	public Client(String address, int port)
	{
		// establish a connection
		try
		{
			socket = new Socket(address, port);
			System.out.println("Connected");

			// takes input from terminal
			input = new DataInputStream(System.in);
```

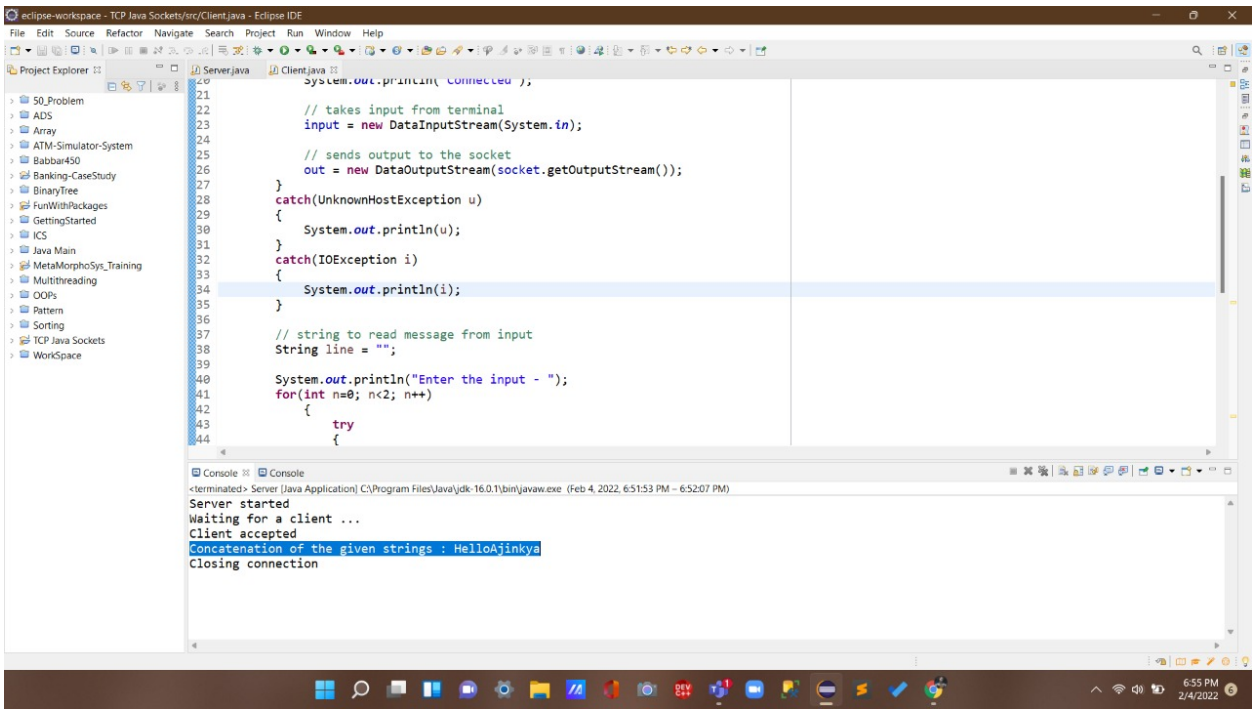
```
        // sends output to the socket
        out = new DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u)
{
        System.out.println(u);
}
catch(IOException i)
{
        System.out.println(i);
}

// string to read message from input
String line = "";

System.out.println("Enter the input - ");
for(int n=0; n<2; n++)
        {
                try
                {
                        System.out.print("String " + (n+1) + " : ");
                        line = input.readLine();
                        out.writeUTF(line);
                }
                catch(IOException i)
                {
                        System.out.println(i);
                }
        }

// close the connection
try
{
        input.close();
        out.close();
        socket.close();
}
catch(IOException i)
{
```

```
                System.out.println(i);
            }
        }

        public static void main(String args[])
        {
            Client client = new Client("127.0.0.1", 5000);
        }
}
```

Client Console:



Server Console:



Conclusion: Developed distributed application through implementing client-server communication programs based on TCP Java Sockets.
