S13/NL/61,63

Assignment 8

Aim: Socket programming with java TCP Client, TCP sever.

LO Mapping: Assignment matches LO4.

A socket is a communications connection point (endpoint) that you can name and address in a network. Socket programming shows how to use socket APIs to establish communication links between remote and local processes.

The processes that use a socket can reside on the same system or different systems on different networks. Sockets are useful for both stand-alone and network applications. Sockets allow you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and they easily allow access to centralized data. Socket application program interfaces are the network standard for TCP/IP.

Server Code

```
import java.io.*;
import java.net.*;


public class Server2 {
    public static void main(String[] args) {
        ServerSocket serverSocket = null;
        Socket clientSocket = null;
        PrintWriter out = null;
        BufferedReader in = null;

        try {
            // Server listens on port 1234
```

```
serverSocket = new ServerSocket(1234);
System.out.println("Server is waiting for a connection...");

// Accept the connection from client
clientSocket = serverSocket.accept();
System.out.println("Client connected: " + clientSocket.getInetAddress());

// Setup input and output streams
in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
out = new PrintWriter(clientSocket.getOutputStream(), true);

// Read numbers and operation from client
String message;
while ((message = in.readLine()) != null) {
    String[] messageParts = message.split(" ");

    if (messageParts.length == 3) {
        double num1 = Double.parseDouble(messageParts[0]);
        double num2 = Double.parseDouble(messageParts[1]);
        String operation = messageParts[2];

        double result = 0;
        boolean validOperation = true;

        // Perform the operation
        switch (operation.toLowerCase()) {
            case "add":
                result = num1 + num2;
                break;
```

```
                case "subtract":
                    result = num1 - num2;
                    break;
                case "multiply":
                    result = num1 * num2;
                    break;
                case "divide":
                    if (num2 != 0) {
                        result = num1 / num2;
                    } else {
                        out.println("Error: Division by zero.");
                        validOperation = false;
                    }
                    break;
                default:
                    out.println("Error: Invalid operation.");
                    validOperation = false;
            }

            // Send the result back to the client
            if (validOperation) {
                out.println("Result: " + result);
            }

        } else {
            out.println("Error: Invalid input. Please send two numbers and an operation.");
        }
    }
```

```
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        } finally {
            try {
                if (in != null) in.close();
                if (out != null) out.close();
                if (clientSocket != null) clientSocket.close();
                if (serverSocket != null) serverSocket.close();
            } catch (IOException e) {
                System.err.println("Error closing resources: " + e.getMessage());
            }
        }
    }
}
```

Client Code

```
import java.io.*;
import java.net.*;

public class Client2 {
    public static void main(String[] args) {
        Socket socket = null;
        PrintWriter out = null;
        BufferedReader in = null;
        BufferedReader userInput = null;

        try {
            // Connect to the server at localhost on port 1234
            socket = new Socket("localhost", 1234);
            System.out.println("Connected to server");
```

```
// Setup input and output streams
out = new PrintWriter(socket.getOutputStream(), true);
in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
userInput = new BufferedReader(new InputStreamReader(System.in));

// Prompt the user for the two numbers and the operation
String userMessage;
String response;

while (true) {
    // Get first number, second number, and operation from the user
    System.out.print("Enter first number: ");
    double num1 = Double.parseDouble(userInput.readLine());

    System.out.print("Enter second number: ");
    double num2 = Double.parseDouble(userInput.readLine());

    System.out.print("Enter operation (add, subtract, multiply, divide): ");
    String operation = userInput.readLine();

    // Send the input to the server
    out.println(num1 + " " + num2 + " " + operation);

    // Receive the result from the server
    String serverResponse = in.readLine();
    System.out.println(serverResponse);

    // Ask the user if they want to perform another operation
```

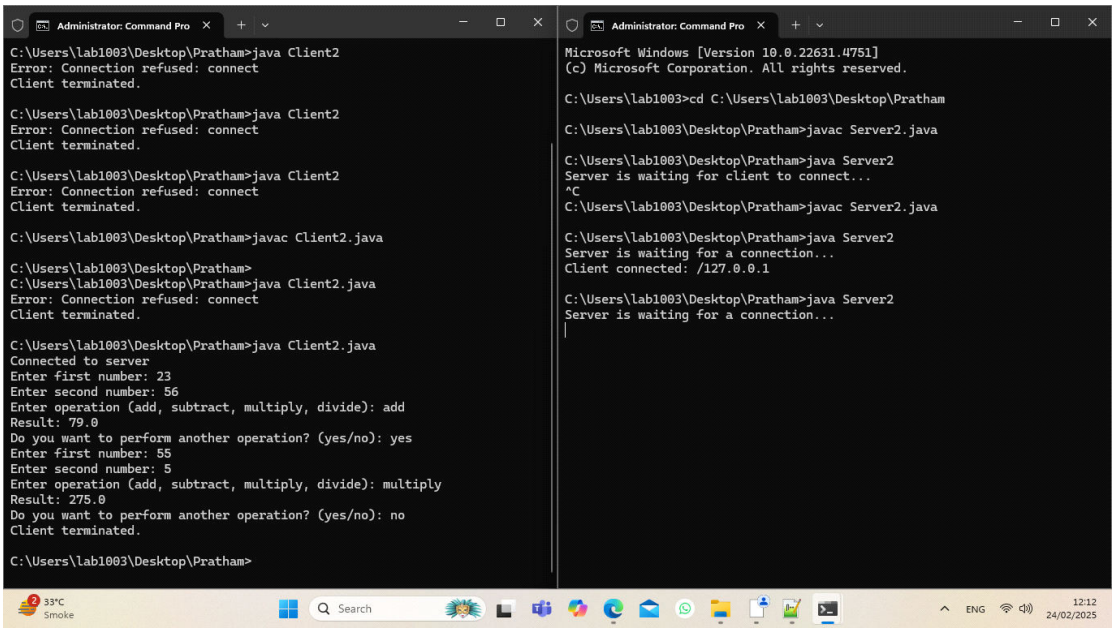
```
                System.out.print("Do you want to perform another operation? (yes/no): ");
                response = userInput.readLine();

                if (response.equalsIgnoreCase("no")) {
                    break; // Exit the loop if the user types "no"
                }
            }

        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        } finally {
            // Ensure resources are closed properly before exiting
            try {
                if (in != null) in.close();
                if (out != null) out.close();
                if (socket != null) socket.close();
                if (userInput != null) userInput.close();
            } catch (IOException e) {
                System.err.println("Error closing resources: " + e.getMessage());
            }
        }

        System.out.println("Client terminated.");
    }
}
```

Output


Administrator: Command Pro
C:\Users\lab1003\Desktop\Pratham>java Client2
Error: Connection refused: connect
Client terminated.
C:\Users\lab1003\Desktop\Pratham>java Client2
Error: Connection refused: connect
Client terminated.
C:\Users\lab1003\Desktop\Pratham>java Client2
Error: Connection refused: connect
Client terminated.
C:\Users\lab1003\Desktop\Pratham>javac Client2.java
C:\Users\lab1003\Desktop\Pratham>
C:\Users\lab1003\Desktop\Pratham>java Client2.java
Error: Connection refused: connect
Client terminated.
C:\Users\lab1003\Desktop\Pratham>java Client2.java
Connected to server
Enter first number: 23
Enter second number: 56
Enter operation (add, subtract, multiply, divide): add
Result: 79.0
Do you want to perform another operation? (yes/no): yes
Enter first number: 55
Enter second number: 5
Enter operation (add, subtract, multiply, divide): multiply
Result: 275.0
Do you want to perform another operation? (yes/no): no
Client terminated.
C:\Users\lab1003\Desktop\Pratham>
Administrator: Command Pro
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.
C:\Users\lab1003>cd C:\Users\lab1003\Desktop\Pratham
C:\Users\lab1003\Desktop\Pratham>javac Server2.java
C:\Users\lab1003\Desktop\Pratham>java Server2
Server is waiting for client to connect...
^C
C:\Users\lab1003\Desktop\Pratham>javac Server2.java
C:\Users\lab1003\Desktop\Pratham>java Server2
Server is waiting for a connection...
Client connected: /127.0.0.1
C:\Users\lab1003\Desktop\Pratham>java Server2
Server is waiting for a connection...
33°C
Smoke
Search
ENG
12:12
24/02/2025