**Assignment 1:**

Write a simple TCP and UDP iterative server and client program in Java to evaluate arithmetic expressions. The client-server system has the following functionalities:

a. The client asks for an arithmetic expression from the user. The user enters an arithmetic expression containing only positive integers and the binary operators +, -, \*, and /. The arithmetic expression can be at most 100 characters long. There may or may not be spaces before and after the operators. For example, all of the following are valid expressions to enter: “13 + 42\*5”, “10+2/4”, “5 + 6 - 3”. Assume that all operators have the same precedence and are left-associative.

b. The client sends the expression to the server. You should send only the number of bytes needed to send the expression properly and not send the complete buffer if not needed.

c. The server computes the expression and sends the integer back to the client.

d. The client displays the result on the screen. It then prompts the user to enter the next expression.

e. The client terminates when the user enters a ‘ –1’.

**Roll Number :** 19CS4151

**SUBMITTED BY STUDENT NAME : VIPLAV PRADEEP PATIL**

**Ex.No: 1:** **Implementation of Client-Server Communication Using TCP**

**AIM:**

To implement a server and client in java using TCP sockets and UDP Sockets for arithmetic calculation.

**DESCRIPTION:**

TCP Clients sends a request to the server and the server will receive the request and calculates the Expression Every time client communicates with the server and receives a result of expression from it.

**ALGORITHM:**

Server

1. Create a server socket and bind it to port.

2. Listen for new connection and when a connection arrives, accept it.

3. Read the Client's message and display it

4. Get a message from the user and send it to client

5. Repeat steps 3-4 until the client sends "end"

6. Close all streams

7. Close the server and client socket

8. Stop

Client

1. Create a client socket and connect it to the server‟s port number

2. Get a Expression from the user and send it to server

3. Read the server's result response and display it

4. Repeat steps 2-3 until chat is terminated with "end" message

5. Close all input/output streams

6. Close the client socket

7. Stop

**A)USING TCP**

**PROGRAM:**

*//TCP Server*

import java.net.\*;

import java.io.\*;

import javax.script.ScriptEngine;

import javax.script.ScriptEngineManager;

import javax.script.ScriptException;

public class Server{

public static void main(String[] args)throws Exception{

System.out.println("Server Started");

ServerSocket ss = new ServerSocket(5555);

Socket s = ss.accept();

System.out.println("Client connected");

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String input = "";

// Get JavaScript engine

ScriptEngine engine = new ScriptEngineManager().getEngineByExtension("js");

while (!input.equals("-1")) {

input = din.readUTF();

System.out.println("Client says : " + input);

// Evaluate the expression

Object output = engine.eval(input);

String convertedToString = output.toString(); //method 3

//output = br.readLine();

dout.writeUTF(convertedToString);

dout.flush(); }

din.close();

s.close();

ss.close(); }

}

*//TCP Client*

// This code contributed by Viplav Patil

import java.net.\*;

import java.io.\*;

public class Client {

public static void main(String[] args) throws Exception {

Socket s = new Socket("localhost",5555);

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String input = "", output = "";

while (!input.equals("-1")) {

input = br.readLine();

dout.writeUTF(input);

dout.flush();

output = din.readUTF();

System.out.println("Server says : " + output);

}

din.close();

s.close();

}

}

**OUTPUT**

**Server**:

$ javac Server.java

$ java Server

Server Started

Client Connected

Client says :15\*54

Client says :-1

**Client**:

$ javac Client.java

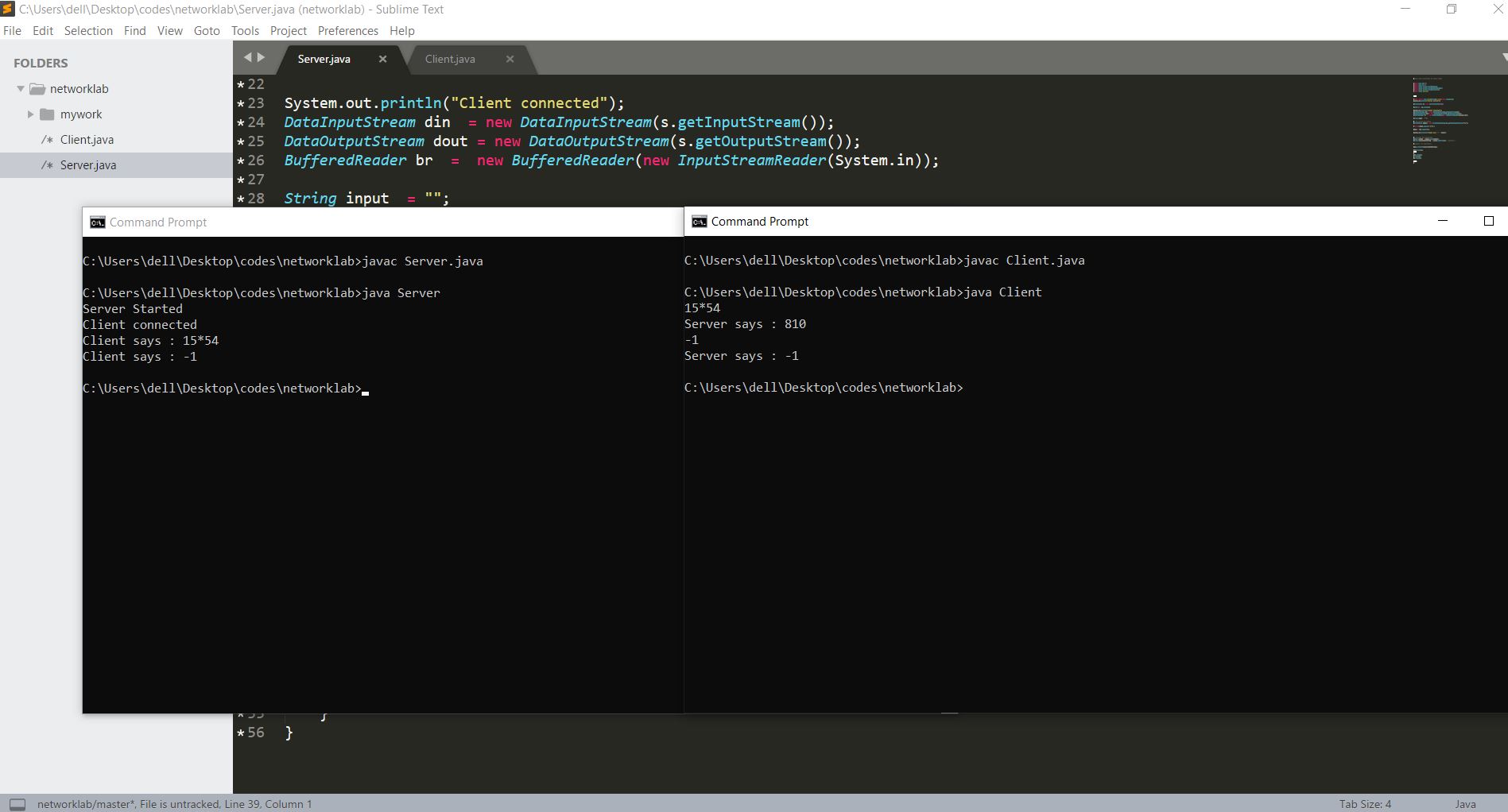
$ java Client

$15\*54

$Server Says :810

$ -1

**RESULT <Screenshots>:**



**B)USING UDP:**

**PROGRAM:**

//UDPSERVER

import java.io.\*;

import java.net.\*;

import javax.script.ScriptEngine;

import javax.script.ScriptEngineManager;

import javax.script.ScriptException;

class UDPServer

{

public static DatagramSocket serversocket;

public static DatagramPacket dp;

public static BufferedReader dis;

public static InetAddress ia;

public static byte buf[] = new byte[1024];

public static int cport = 789,sport=790;

public static void main(String[] a) throws IOException,ScriptException

{

serversocket = new DatagramSocket(sport);

dp = new DatagramPacket(buf,buf.length);

dis = new BufferedReader(new InputStreamReader(System.in));

ia = InetAddress.getLocalHost();

System.out.println("Server is Running...");

// Get JavaScript engine

ScriptEngine engine = new ScriptEngineManager().getEngineByExtension("js");

while(true)

{

serversocket.receive(dp);

String str = new String(dp.getData(), 0,dp.getLength());

if(str.equals("-1"))

{

System.out.println("Terminated...");

break;

}

// Evaluate the expression

Object output = engine.eval(str);

String convertedToString = output.toString(); //method 3

System.out.println("Client: " + str);

//String str1 = new String(dis.readLine());

buf = convertedToString.getBytes();

serversocket.send(new DatagramPacket(buf,convertedToString.length(), ia, cport));

}

}

}

//UDPCLIENT

import java.io.\*;

import java.net.\*;

class UDPClient

{

public static DatagramSocket clientsocket;

public static DatagramPacket dp;

public static BufferedReader dis;

public static InetAddress ia;

public static byte buf[] = new byte[1024];

public static int cport = 789, sport = 790;

public static void main(String[] a) throws IOException

{

clientsocket = new DatagramSocket(cport);

dp = new DatagramPacket(buf, buf.length);

dis = new BufferedReader(new InputStreamReader(System.in));

ia = InetAddress.getLocalHost();

System.out.println("Client is Running... Type -1 to Quit");

while(true)

{

System.out.println("Give Input Expression:");

String str = new String(dis.readLine());

buf = str.getBytes();

if(str.equals("-1"))

{

System.out.println("Terminated...");

clientsocket.send(new DatagramPacket(buf,str.length(), ia,sport));

break;

}

clientsocket.send(new DatagramPacket(buf,str.length(), ia, sport));

clientsocket.receive(dp);

String str2 = new String(dp.getData(), 0,

dp.getLength());

System.out.println("Server: " + str2);

}

}

}

**OUTPUT:**

**Server**:

$ javac UDPServer.java

$ java UDPServer

Server is Running…

Client :5\*2

Client :12+45

Terminated..

**Client**:

$ javac UDPClient.java

$ java UDPClient

$Give Input Expression:

5\*2

$Server :10

$Give Input Expression :

12+45

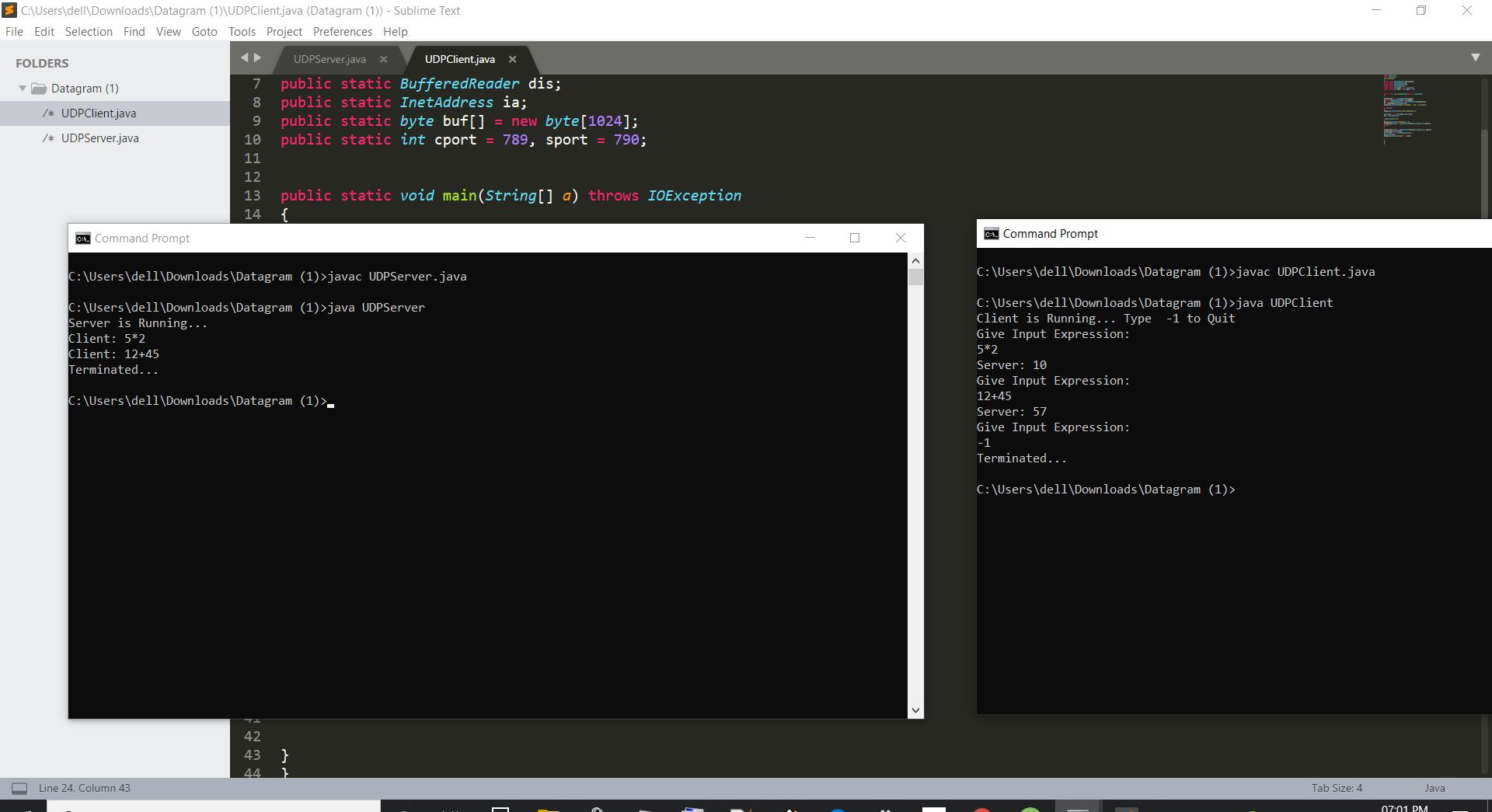
$Server :57

$Give Input Expression :

-1

Terminated…

**RESULT <Screenshots>**

****