DA
Lab Assignment - 4
U18CO018
Shubham Shekhaliya

Implement echo client-server message passing application. Messages sent from the client should be displayed on the server and then the program should terminate.

1. Write a server (TCP) C Program that opens a listening socket and waits to serve client. 2. Write a client (TCP) C Program that connects with the server program knowing IP address and port number.
3. Get the input string from console on client and send it to server, server displays the same string.

Client.java

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

public class Client {
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;

    public Client(String address, int port) {
        // establish a connection
        try {
            socket = new Socket(address, port);
            System.out.println("Connected");
            input = new DataInputStream(System.in);
            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 = "";
        // keep reading until "Over" is input
        while (!line.equals("Over")) {
```

```
            try {
                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);
    }
}
```

Server.java

```
import java.io.BufferedInputStream;
import java.io.DataInputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    private Socket socket;

    private Server(int port) {
        // Create a new server socket
        ServerSocket serverSocket = null;
        try {
            serverSocket = new ServerSocket(port);
            System.out.println("Server is running on port " + port);
            // Wait for a client to connect
            socket = serverSocket.accept();
            System.out.println("Client connected");
            DataInputStream in = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));
            String line = "";
            // reads message from client until "Over" is sent
            while (!line.equals("Over")) {
                try {
```

```
                    line = in.readUTF();
                    System.out.println(line);
                } catch (IOException i) {
                    System.out.println(i);
                }
            }
            System.out.println("Closing connection");
            // close connection
            socket.close();
            in.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        int port = 5000;
        // Create a new server socket
        Server server = new Server(port);
    }
}
```

Client

```
Connected
Hello
Hi
Client here
Over
```

Server

```
Server is running on port 5000
Client connected
Hello
Hi
Client here
Over
Closing connection
```