

WEEK-9

Write a program to implement Client - Server communication for chat using Transmission Control Protocol (TCP).

Server.java

```
import java.net.*;
import java.io.*;
public class Server {
    public static void main(String[] args) throws Exception {
        System.out.println("server is connected");
        ServerSocket ss=new ServerSocket(3333);
        System.out.println("Server is waiting for client request");
        Socket s=ss.accept();
        System.out.println("Client is connected, start chatting");

        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        String str="",str2="";
        while(!str.equals("stop"))
        {
            str=din.readUTF();
            System.out.println("Client Says : "+str);
            str2=br.readLine();
            dout.writeUTF(str2);
            dout.flush();
        }
        din.close();
        ss.close();
    }
}
```

Client.java

```
import java.net.*;
import java.io.*;
public class Client {
    public static void main(String[] args) throws Exception {
        try {
            Socket s=new Socket("localhost",3333);
            System.out.println("Start Chatting...");
            DataInputStream din=new DataInputStream(s.getInputStream());
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

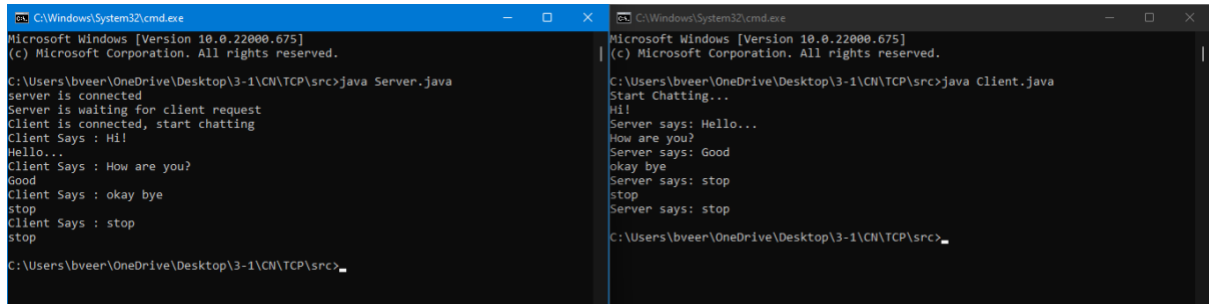
            String str="",str2="";
            while(!str.equals("stop"))
            {
                str=br.readLine();
                dout.writeUTF(str);
                dout.flush();
                str2=din.readUTF();
                System.out.println("Server says: "+str2);
            }
            dout.close();
        }
    }
}
```

```

        s.close();
    }
    catch (ConnectException e) {
        System.out.println("Server is offline");
    }
}
}

```

OUTPUT



The image shows two side-by-side Windows command prompt windows. The left window is titled 'C:\Windows\System32\cmd.exe' and shows the execution of 'java Server.java'. The output indicates the server is connected, waiting for a client request, and then receives a connection. It prints 'Hello...' and 'Client Says : Hi!'. The client then sends 'How are you?', 'Good', 'okay bye', and 'stop'. The server responds with 'Hello...', 'Good', 'okay bye', and 'stop'. The right window is also titled 'C:\Windows\System32\cmd.exe' and shows the execution of 'java Client.java'. The output shows the client starting a chat, sending 'Hi!', and receiving 'Hello...' from the server. The client then sends 'How are you?' and receives 'Good' from the server. The client sends 'okay bye' and receives 'okay bye' from the server. The client sends 'stop' and receives 'stop' from the server. The client then sends 'stop' and receives 'stop' from the server.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveen\OneDrive\Desktop\3-1\CN\TCP\src>java Server.java
server is connected
Server is waiting for client request
Client is connected, start chatting
Client Says : Hi!
Hello...
Client Says : How are you?
Good
Client Says : okay bye
stop
Client Says : stop
stop
C:\Users\bveen\OneDrive\Desktop\3-1\CN\TCP\src>

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bveen\OneDrive\Desktop\3-1\CN\TCP\src>java Client.java
Start Chatting...
Hi!
Server says: Hello...
How are you?
Server says: Good
okay bye
Server says: stop
stop
Server says: stop
C:\Users\bveen\OneDrive\Desktop\3-1\CN\TCP\src>

```