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