1. Implement chatting application using java socket programming

TCP client –

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

class tcpclient {

    public static void main(String args[]) {
        try {
            Socket s1 = new Socket("localhost", 1520);

            DataInputStream dis;
            DataOutputStream dos;
            InputStream is;
            OutputStream os;

            is = s1.getInputStream();
            os = s1.getOutputStream();
            dis = new DataInputStream(is);
            dos = new DataOutputStream(os);

            Scanner sc = new Scanner(System.in);
            String str;
            do {

                System.out.println("Data is send");
                str = sc.nextLine();
                dos.writeUTF(str);
                System.out.println("Data Received:");
                str = dis.readUTF();
                System.out.println(str);
            } while (!str.equals("STOP"));
        } catch (Exception e) {
        }
    }
}
```

TCP server –

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

class tcpserver
{
public static void main(String args[])
{
try
{
ServerSocket s1 = new ServerSocket(1520);
Socket s2 = s1.accept();

DataInputStream dis;
DataOutputStream dos;

InputStream is;
OutputStream os;

is = s2.getInputStream();
os = s2.getOutputStream();

dis = new DataInputStream(is);
dos = new DataOutputStream(os);

Scanner sc = new Scanner(System.in);

String str;
do
{

System.out.println("Data Received:");
str = dis.readUTF();
System.out.println(str);
System.out.println("Data is send:");
str = sc.nextLine();
dos.writeUTF(str);
}while(!str. equals("STOP"));

}
catch(Exception e)
{
}
}
}
```

2. Develop a TCP client-server application where the client sends two numbers to the server, and the server returns their sum.

TCP client –

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

class tcpclient {

    public static void main(String args[]) {
        try {
            // Connect to the server using its IP address and port
            Socket s1 = new Socket("10.1.4.194", 1520);

            // Set up input/output streams
            DataInputStream dis = new DataInputStream(s1.getInputStream());
            DataOutputStream dos = new DataOutputStream(s1.getOutputStream());

            Scanner sc = new Scanner(System.in);

            // Get two numbers from the user
            System.out.print("Enter first number: ");
            int num1 = sc.nextInt();

            System.out.print("Enter second number: ");
            int num2 = sc.nextInt();

            // Send both numbers to server
            dos.writeInt(num1);
            dos.writeInt(num2);
            System.out.println("Numbers sent to server");

            // Receive and display the sum from server
            int result = dis.readInt();
            System.out.println("Sum received from server: " + result);

            // Close all connections
            dis.close();
            dos.close();
            s1.close();

        } catch (Exception e) {
            e.printStackTrace(); // Print error for debugging
        }
    }
}
```

TCP server -

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

class tcpserver {
    public static void main(String args[]) {
        try {
            ServerSocket serverSocket = new ServerSocket(1520);
            System.out.println("Server is waiting for client...");

            Socket socket = serverSocket.accept();
            System.out.println("Client connected!");

            // Get streams
            DataInputStream dis = new DataInputStream(socket.getInputStream());
            DataOutputStream dos = new DataOutputStream(socket.getOutputStream());

            // Read two integers
            int num1 = dis.readInt();
            int num2 = dis.readInt();

            System.out.println("Received numbers: " + num1 + " and " + num2);

            // Perform sum
            int sum = num1 + num2;
            System.out.println("Sending sum: " + sum);

            // Send result back to client
            dos.writeInt(sum);

            // Close everything
            dis.close();
            dos.close();
            socket.close();
            serverSocket.close();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```