Computer Networks

# Assignment-8

# Name: S. Vishwajith

# Register Number: 23BCE1145

## Question-1 Part-A:

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client\_23BCE1145 {

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream output = null;

    public Client\_23BCE1145(String *address*, *int* *port*){

        try{

            socket = **new** Socket(*address*, *port*);

            System.out.println("Client is connected!\n");

            input = **new** DataInputStream(socket.getInputStream());

            output = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        String[] arr = "Welcome to Computer Networks lab. End".split(" ");

        for(*int* fc = 0; fc < arr.length; fc++){

            String line = arr[fc];

            if(line.equals("End")){

                try{

                    output.writeUTF(fc + " " + line);

                }

                catch(IOException *i*){

                    System.out.println(i);

                }

                break;

            }

            try{

                output.writeUTF(fc + " " + line);

                System.out.println("Frame " + (fc) + " sent.");

                String[] ack;

                while(true){

                    try {

                        socket.setSoTimeout(5000);

                        ack=input.readUTF().split(" ");

                        if (ack[0].equals("Acknowledged") && ack[1].equals(Integer.toString((fc%2)^1))){

                            System.out.println("Acknowledgement " + ((fc%2)^1) + " Received. Sending next frame.");

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Ackowledgement " + ((fc%2)^1) + " not received after 5 seconds. Resending Frame " + (fc+1) + ".");

                        output.writeUTF(fc + " " + line);

                    }

                }

            }

            catch(IOException *i*){

                System.out.println(i);

            }

        }

        System.out.println("\nClosing Connection.");

        try{

            input.close();

            output.close();

            socket.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Client\_23BCE1145 client = **new** Client\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Server:

import java.io.*\**;

import java.net.*\**;

public class Server\_23BCE1145{

    private ServerSocket server = null;

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream output = null;

    public Server\_23BCE1145(*int* *port*){

        try{

            System.out.println("Starting server.");

            server = **new** ServerSocket(*port*);

            System.out.println("Waiting for a connection.");

            socket = server.accept();

            System.out.println("Connection with client established.\n");

            input = **new** DataInputStream(socket.getInputStream());

            output = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        String line = "";

*int* frameCount = 0;

        while(!line.equals("End")){

            try{

                String[] arr = input.readUTF().split(" ");

                line = arr[1];

*int* fc = Integer.parseInt(arr[0]);

                if(line.equals("End")){

                    break;

                }

                System.out.println("Frame " + frameCount + " received. Message is: " + line);

                frameCount++;

                output.writeUTF("Acknowledged " + ((fc%2)^1));

            }

            catch(IOException *i*){

                System.out.println(i);

            }

        }

        System.out.println("\nClosing Connection.");

        try{

            input.close();

            output.close();

            socket.close();

            server.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Server\_23BCE1145 server = **new** Server\_23BCE1145(5000);

        server.Work();

    }

}

### Outputs:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Question-1 Part-B:

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client\_23BCE1145 {

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream output = null;

    public Client\_23BCE1145(String *address*, *int* *port*){

        try{

            socket = **new** Socket(*address*, *port*);

            System.out.println("Client is connected!\n");

            input = **new** DataInputStream(socket.getInputStream());

            output = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        String[] arr = "Welcome to Computer Networks lab. End".split(" ");

        for(*int* fc = 0; fc < arr.length; fc++){

            String line = arr[fc];

            if(line.equals("End")){

                try{

                    output.writeUTF(fc + " " + line);

                }

                catch(IOException *i*){

                    System.out.println(i);

                }

                break;

            }

            try{

                output.writeUTF(fc + " " + line);

                System.out.println("Frame " + (fc) + " sent.");

                String[] ack;

                while(true){

                    try {

                        socket.setSoTimeout(5000);

                        ack=input.readUTF().split(" ");

                        if (ack[0].equals("Acknowledged") && ack[1].equals(Integer.toString((fc%2)^1))){

                            System.out.println("Acknowledgement " + ((fc%2)^1) +" Received. Sending next frame.");

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Ackowledgement " + ((fc%2)^1) + " not received after 5 seconds. Resending Frame.");

                        output.writeUTF(fc + " " + line);

                    }

                }

            }

            catch(IOException *i*){

                System.out.println(i);

            }

        }

        System.out.println("\nClosing Connection.");

        try{

            input.close();

            output.close();

            socket.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Client\_23BCE1145 client = **new** Client\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Server:

import java.io.*\**;

import java.net.*\**;

public class Server\_23BCE1145{

    private ServerSocket server = null;

    private Socket socket = null;

    private DataInputStream input = null;

    private DataOutputStream output = null;

    public Server\_23BCE1145(*int* *port*){

        try{

            System.out.println("Starting server.\n");

            server = **new** ServerSocket(*port*);

            System.out.println("Waiting for a connection.\n");

            socket = server.accept();

            System.out.println("Connection with client established.\n");

            input = **new** DataInputStream(socket.getInputStream());

            output = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        String line = "";

*int* frameCount = 0;

        while(!line.equals("End")){

            try{

                String[] arr = input.readUTF().split(" ");

*int* fc = Integer.parseInt(arr[0]);

                line = arr[1];

                if(line.equals("End")){

                    break;

                }

                if (frameCount==3) {

                    output.writeUTF("Not Acknowledged");

                    System.out.println("Acknowledgement not sent.");

                }

                else {

                    output.writeUTF("Acknowledged " + ((fc%2)^1));

                    System.out.println("Acknowledgement " + ((fc%2)^1) + " sent.");

                }

                System.out.println("Frame " + frameCount + " received. Message is: " + line);

                frameCount++;

            }

            catch(IOException *i*){

                System.out.println(i);

            }

        }

        System.out.println("\nClosing Connection.");

        try{

            input.close();

            output.close();

            socket.close();

            server.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Server\_23BCE1145 server = **new** Server\_23BCE1145(5000);

        server.Work();

    }

}

### Outputs:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Question-2 Part-A:

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client\_23BCE1145 {

    private Socket socket = null;

    private DataInputStream in = null;

    private DataOutputStream out = null;

    public Client\_23BCE1145(String *address*, *int* *port*) {

        try {

            System.out.println("\nConnecting to the server...");

            socket = **new** Socket(*address*, *port*);

            System.out.println("Connected.\n");

            in = **new** DataInputStream(socket.getInputStream());

            out = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        try{

            String[] data = "Networks lab is easy to understand and implement. End".split(" ");

*int* n = data.length;

*int* frameSize = 4, i, j;

            for(i = 0; i < frameSize; i++){

                out.writeUTF(i + " " + data[i]);

            }

            for(i = frameSize; i < n; i++){

                if(data[i].equalsIgnoreCase("End")){

                    out.writeUTF(i + " " + data[i]);

                    break;

                }

                while(true){

                    try{

                        socket.setSoTimeout(3000);

                        String[] ack = in.readUTF().split(" ");

                        if(ack[0].equalsIgnoreCase("Acknowledged") && Integer.parseInt(ack[1]) >= (i - frameSize)){

                            System.out.println("Acknowledgement " + ack[1] + " received. Sending next frame.");

                            for(j = i; j <= Integer.parseInt(ack[1])+frameSize; j++){

                                out.writeUTF(j + " " + data[j]);

                            }

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Acknowledgement " + (i - frameSize) + " not received after 3 seconds. Resending Frame " + (i - frameSize + 1) + ".");

                        for(j = i - frameSize; j < i; j++){

                            out.writeUTF(j + " " + data[j]);

                        }

                    }

                }

            }

            for(i = frameSize; i < n-1; i++){

                while(true){

                    try{

                        socket.setSoTimeout(3000);

                        String[] ack = in.readUTF().split(" ");

                        if(ack[0].equalsIgnoreCase("Acknowledged") && Integer.parseInt(ack[1]) >= i){

                            System.out.println("Acknowledgement " + ack[1] + " received.");

                            for(j = i; j < 8; j++){

                                out.writeUTF(j + " " + data[j]);

                            }

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Acknowledgement " + i + " not received after 3 seconds. Resending Frame " + (i - frameSize + 1) + ".");

                        for(j = i; j < n-1; j++){

                            out.writeUTF(j + " " + data[j]);

                        }

                    }

                }

            }

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("\nClosing connection...");

            in.close();

            out.close();

            socket.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*) {

        Client\_23BCE1145 client = **new** Client\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Server:

import java.io.*\**;

import java.net.*\**;

public class Server\_23BCE1145{

    private Socket socket = null;

    private ServerSocket server = null;

    private DataInputStream in = null;

    private DataOutputStream out = null;

    public Server\_23BCE1145(*int* *port*){

        try{

            System.out.println("\nStarting the server...");

            server = **new** ServerSocket(*port*);

            System.out.println("Waiting for a client ...");

            socket = server.accept();

            System.out.println("Client accepted.\n");

            in = **new** DataInputStream(socket.getInputStream());

            out = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        try{

            String[] data = **new** String[9];

            while(true){

                String[] frame = in.readUTF().split(" ");

                if(frame[1].equalsIgnoreCase("End")){

                    data[Integer.parseInt(frame[0])] = frame[1];

                    break;

                }

                data[Integer.parseInt(frame[0])] = frame[1];

                System.out.println("Frame " + frame[0] + " received. Data is: \"" + frame[1] + "\".\nSending Acknowledgement " + frame[0] + ".");

                out.writeUTF("Acknowledged " + frame[0]);

            }

            System.out.print("\nThe complete data is: ");

            for(*int* j = 0; j < 8; j++){

                System.out.print(data[j] + " ");

            }

            System.out.println();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("\nClosing connection...");

            in.close();

            out.close();

            socket.close();

            server.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Server\_23BCE1145 server = **new** Server\_23BCE1145(5000);

        server.Work();

    }

}

### Outputs:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Question-2 Part-B:

### Server:

import java.io.*\**;

import java.net.*\**;

public class Server\_23BCE1145{

    private Socket socket = null;

    private ServerSocket server = null;

    private DataInputStream in = null;

    private DataOutputStream out = null;

    public Server\_23BCE1145(*int* *port*){

        try{

            System.out.println("\nStarting the server...");

            server = **new** ServerSocket(*port*);

            System.out.println("Waiting for a client ...");

            socket = server.accept();

            System.out.println("Client accepted.\n");

            in = **new** DataInputStream(socket.getInputStream());

            out = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        try{

            String[] data = **new** String[9];

            while(true){

                String[] frame = in.readUTF().split(" ");

                if(frame[1].equalsIgnoreCase("End")){

                    data[Integer.parseInt(frame[0])] = frame[1];

                    out.writeUTF("Acknowledged " + frame[0]);

                    break;

                }

                data[Integer.parseInt(frame[0])] = frame[1];

                if(Integer.parseInt(frame[0]) == 4){

                    System.out.println("Frame " + frame[0] + " received. Data is: \"" + frame[1] + "\".\nNot sending Acknowledgement.");

                    out.writeUTF("Not Acknowledged frame.");

                }

                else{

                    if(frame[1].equalsIgnoreCase("Corrupted")){

                        System.out.println("Frame " + frame[0] + " received. Data is: \"" + frame[1] + "\".\nNot sending Acknowledgement.");

                        out.writeUTF("Not Acknowledged frame.");

                    }

                    else{

                        System.out.println("Frame " + frame[0] + " received. Data is: \"" + frame[1] + "\".\nSending Acknowledgement.");

                        out.writeUTF("Acknowledged " + frame[0]);

                    }

                }

            }

            System.out.print("\nThe complete data is: ");

            for(*int* j = 0; j < 8; j++){

                System.out.print(data[j] + " ");

            }

            System.out.println();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("\nClosing connection...");

            in.close();

            out.close();

            socket.close();

            server.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*){

        Server\_23BCE1145 server = **new** Server\_23BCE1145(5000);

        server.Work();

    }

}

### Client:

import java.io.*\**;

import java.net.*\**;

public class Client\_23BCE1145 {

    private Socket socket = null;

    private DataInputStream in = null;

    private DataOutputStream out = null;

    public Client\_23BCE1145(String *address*, *int* *port*) {

        try {

            System.out.println("\nConnecting to the server...");

            socket = **new** Socket(*address*, *port*);

            System.out.println("Connected.\n");

            in = **new** DataInputStream(socket.getInputStream());

            out = **new** DataOutputStream(socket.getOutputStream());

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        try{

            String[] data = "Networks lab is easy to understand and implement. End".split(" ");

*int* n = data.length;

*int* frameSize = 4, i, j;

            for(i = 0; i < frameSize; i++){

                out.writeUTF(i + " " + data[i]);

            }

            for(i = frameSize; i < n; i++){

                if(data[i].equalsIgnoreCase("End")){

                    out.writeUTF(i + " " + data[i]);

                    break;

                }

                while(true){

                    try{

                        socket.setSoTimeout(3000);

                        String[] ack = in.readUTF().split(" ");

                        if(ack[0].equalsIgnoreCase("Acknowledged") && Integer.parseInt(ack[1]) >= (i - frameSize)){

                            System.out.println("Acknowledgement " + ack[1] + " received. Sending next frame.");

                            for(j = i; j <= Integer.parseInt(ack[1])+frameSize; j++){

                                out.writeUTF(j + " " + data[j]);

                            }

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Acknowledgement " + (i - frameSize) + " not received after 3 seconds. Resending Frame " + (i - frameSize + 1) + ".");

                        for(j = i - frameSize; j < i; j++){

                            out.writeUTF(j + " " + data[j]);

                        }

                    }

                }

            }

            i = frameSize;

            while(true){

*boolean* flag = false;

                while(true){

                    try{

                        socket.setSoTimeout(3000);

                        String[] ack = in.readUTF().split(" ");

                        if(ack[0].equalsIgnoreCase("Acknowledged")){

                            i = Integer.parseInt(ack[1]);

                            if(i == n-1){

                                flag = true;

                                break;

                            }

                            System.out.println("Acknowledgement " + ack[1] + " received.");

                            break;

                        }

                    }

                    catch(SocketTimeoutException *ste*){

                        System.out.println("Acknowledgement " + i + " not received after 3 seconds. Resending Frame " + (i - frameSize + 1) + ".");

                        for(j = i; j < n-1; j++){

                            out.writeUTF(j + " " + data[j]);

                        }

                    }

                }

                if(flag){

                    break;

                }

            }

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("\nClosing connection...");

            in.close();

            out.close();

            socket.close();

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public static *void* main(String[] *args*) {

        Client\_23BCE1145 client = **new** Client\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Outputs:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.