Computer Networks

# Assignment-7

# Name: S. Vishwajith

# Register Number: 23BCE1145

## Question-1:

### Server:

import java.io.*\**;

import java.net.*\**;

public class Server\_23BCE1145 {

    private ServerSocket server = null;

    private Socket socket = null;

    private DataInputStream input = null;

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

        try{

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

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

        try{

*int* srn = Integer.parseInt(input.readUTF());

*int* cc = Integer.parseInt(input.readUTF());

*int* mo = Integer.parseInt(input.readUTF());

*int* ap = Integer.parseInt(input.readUTF());

*int* res1 = Integer.parseInt(input.readUTF());

*int* tot = srn + cc + mo + ap, bits = 12;

*int* rem = (tot >> bits) & 0b11;

*int* rem2 = tot & 0xFFF;

*int* res2 = ~(rem + rem2 + res1) & 0xFFF;

            System.out.println("Student Register Number: " + srn + ".");

            System.out.println("Course Code: " + cc + ".");

            System.out.println("Marks Obtained: " + mo + ".");

            System.out.println("Attendance Percentage: " + ap + ".\n");

            if(res2 == 0){

                System.out.println("Checksum verification passed.\n");

            }

            else{

                System.out.println("Checksum verficiation failed.\n");

            }

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("Closing connection.");

            input.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 DataOutputStream output = null;

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

        try{

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

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

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

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        }

        catch(IOException *i*){

            System.out.println(i);

        }

    }

    public *void* Work(){

*int* srn = 8942, cc = 1308, mo = 85, ap = 92;

*int* tot = srn + cc + mo + ap, bits = 12;

*int* rem = (tot >> bits) & 0b11;

*int* rem2 = tot & 0xFFF;

*int* res = ~(rem + rem2) & 0xFFF;

        try{

            output.writeUTF(""+srn);

            output.writeUTF("1318");

            output.writeUTF(""+mo);

            output.writeUTF(""+ap);

            output.writeUTF(""+res);

            System.out.println("Student Register Number: " + srn + ".");

            System.out.println("Course Code: " + cc + ".");

            System.out.println("Marks Obtained: " + mo + ".");

            System.out.println("Attendance Percentage: " + ap + ".\n");

            System.out.println("Calculated checksum is: " + res + ".\n");

        }

        catch(IOException *i*){

            System.out.println(i);

        }

        try{

            System.out.println("Closing connection.");

            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();

    }

}

### 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:

### Server:

import java.io.*\**;

import java.net.*\**;

import java.util.*\**;

public class Server\_23BCE1145 {

    private ServerSocket server = null;

    private Socket socket1 = null, socket2 = null, socket3 = null;

    private ObjectInputStream input1 = null, input2 = null, input3 = null;

    private DataOutputStream output1 = null, output2 = null, output3 = null;

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

        try {

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

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

            System.out.println("Waiting for 3 clients...\n");

            socket1 = server.accept();

            System.out.println("Client 1 connected.");

            socket2 = server.accept();

            System.out.println("Client 2 connected.");

            socket3 = server.accept();

            System.out.println("Client 3 connected.\n");

            input1 = **new** ObjectInputStream(socket1.getInputStream());

            input2 = **new** ObjectInputStream(socket2.getInputStream());

            input3 = **new** ObjectInputStream(socket3.getInputStream());

            output1 = **new** DataOutputStream(socket1.getOutputStream());

            output2 = **new** DataOutputStream(socket2.getOutputStream());

            output3 = **new** DataOutputStream(socket3.getOutputStream());

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    @SuppressWarnings("unchecked")

    public *void* Work() {

        try {

*Set*<String> words1 = (*Set*<String>) input1.readObject();

*Set*<String> words2 = (*Set*<String>) input2.readObject();

*Set*<String> words3 = (*Set*<String>) input3.readObject();

*double* sim12 = calculateJaccardSimilarity(words1, words2);

*double* sim13 = calculateJaccardSimilarity(words1, words3);

*double* sim23 = calculateJaccardSimilarity(words2, words3);

*double* avgSimilarity = (sim12 + sim13 + sim23) / 3.0 \* 100;

            sendResult(output1, avgSimilarity);

            sendResult(output2, avgSimilarity);

            sendResult(output3, avgSimilarity);

            System.out.println("Similarity sent to all clients.\n");

        }

        catch (IOException | ClassNotFoundException *i*) {

            System.out.println(i);

        }

        closeConnections();

    }

    private *double* calculateJaccardSimilarity(*Set*<String> *set1*, *Set*<String> *set2*) {

*Set*<String> intersection = **new** HashSet<>(*set1*);

        intersection.retainAll(*set2*);

*Set*<String> union = **new** HashSet<>(*set1*);

        union.addAll(*set2*);

        return (*double*) intersection.size() / union.size();

    }

    private *void* sendResult(DataOutputStream *output*, *double* *similarity*) throws IOException {

*output*.writeUTF(String.format("%.2f", *similarity*));

*output*.flush();

    }

    private *void* closeConnections() {

        try {

            System.out.println("Closing connections.");

            input1.close();

            input2.close();

            input3.close();

            output1.close();

            output2.close();

            output3.close();

            socket1.close();

            socket2.close();

            socket3.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 1:

import java.io.*\**;

import java.net.*\**;

import java.util.*\**;

public class Client1\_23BCE1145 {

    private Socket socket = null;

    private ObjectOutputStream output = null;

    private DataInputStream input = null;

    public Client1\_23BCE1145(String *address*, *int* *port*) {

        try {

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

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

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

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

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    public *void* Work() {

        try {

            File file = **new** File("file1.txt");

*Set*<String> words = getWordsFromFile(file);

            output.writeObject(words);

            output.flush();

            String similarity = input.readUTF();

            System.out.println("Received Similarity: " + similarity + "%\n");

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

        closeConnection();

    }

    private *Set*<String> getWordsFromFile(File *file*) throws IOException {

        BufferedReader reader = **new** BufferedReader(**new** FileReader(*file*));

*Set*<String> words = **new** HashSet<>();

        String line;

        while ((line = reader.readLine()) != null) {

            String[] tokens = line.toLowerCase().split("\\W+");

            words.addAll(Arrays.asList(tokens));

        }

        reader.close();

        return words;

    }

    private *void* closeConnection() {

        try {

            System.out.println("Closing connection.");

            output.close();

            input.close();

            socket.close();

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

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

        Client1\_23BCE1145 client = **new** Client1\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Client 2:

import java.io.*\**;

import java.net.*\**;

import java.util.*\**;

public class Client2\_23BCE1145 {

    private Socket socket = null;

    private ObjectOutputStream output = null;

    private DataInputStream input = null;

    public Client2\_23BCE1145(String *address*, *int* *port*) {

        try {

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

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

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

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

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    public *void* Work() {

        try {

            File file = **new** File("file2.txt");

*Set*<String> words = getWordsFromFile(file);

            output.writeObject(words);

            output.flush();

            String similarity = input.readUTF();

            System.out.println("Received Similarity: " + similarity + "%\n");

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

        closeConnection();

    }

    private *Set*<String> getWordsFromFile(File *file*) throws IOException {

        BufferedReader reader = **new** BufferedReader(**new** FileReader(*file*));

*Set*<String> words = **new** HashSet<>();

        String line;

        while ((line = reader.readLine()) != null) {

            String[] tokens = line.toLowerCase().split("\\W+");

            words.addAll(Arrays.asList(tokens));

        }

        reader.close();

        return words;

    }

    private *void* closeConnection() {

        try {

            System.out.println("Closing connection.");

            output.close();

            input.close();

            socket.close();

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

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

        Client2\_23BCE1145 client = **new** Client2\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Client 3:

import java.io.*\**;

import java.net.*\**;

import java.util.*\**;

public class Client3\_23BCE1145 {

    private Socket socket = null;

    private ObjectOutputStream output = null;

    private DataInputStream input = null;

    public Client3\_23BCE1145(String *address*, *int* *port*) {

        try {

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

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

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

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

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

    public *void* Work() {

        try {

            File file = **new** File("file3.txt");

*Set*<String> words = getWordsFromFile(file);

            output.writeObject(words);

            output.flush();

            String similarity = input.readUTF();

            System.out.println("Received Similarity: " + similarity + "%\n");

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

        closeConnection();

    }

    private *Set*<String> getWordsFromFile(File *file*) throws IOException {

        BufferedReader reader = **new** BufferedReader(**new** FileReader(*file*));

*Set*<String> words = **new** HashSet<>();

        String line;

        while ((line = reader.readLine()) != null) {

            String[] tokens = line.toLowerCase().split("\\W+");

            words.addAll(Arrays.asList(tokens));

        }

        reader.close();

        return words;

    }

    private *void* closeConnection() {

        try {

            System.out.println("Closing connection.");

            output.close();

            input.close();

            socket.close();

        }

        catch (IOException *i*) {

            System.out.println(i);

        }

    }

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

        Client3\_23BCE1145 client = **new** Client3\_23BCE1145("127.0.0.1", 5000);

        client.Work();

    }

}

### Files:

A screenshot of a computer program

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

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

AI-generated content may be incorrect.

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

## Question-3:

### Server:

import java.net.*\**;

import java.io.*\**;

public class Server\_23BCE1145{

    private Socket socket=null;

    private ServerSocket server=null;

    private DataInputStream input=null;

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

        try{

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

            System.out.println("Server Started.\nWaiting for a client...");

            socket=server.accept();

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

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

            String line="";

            while (!line.toUpperCase().equalsIgnoreCase("END")){

                try{

                    line=input.readUTF();

                    if (line.equalsIgnoreCase("END")){

                        break;

                    }

                    line=line.substring(0,6)+(Integer.parseInt(line.charAt(6)+"")^1)+line.substring(7,line.length());

                    System.out.println("Received Codeword: "+line);

*int* len=line.length();

                    String par="";

                    line=**new** StringBuilder(line).reverse().toString();

                    for (*int* i=0;(*int*)Math.pow(2,i)<len;i++){

*int* pa=parity(line,i);

                        System.out.println("P"+(*int*)Math.pow(2,i)+": "+pa);

                        par=pa+par;

                    }

*int* bin=Integer.parseInt(par,2);

                    if (bin==0){

                        System.out.println("No error detected");

                    }

                    else{

                        System.out.println("The bit number "+bin+" is an error and thus needs to be flipped");

                        line=line.substring(0,bin-1)+(Integer.parseInt(line.charAt(bin-1)+"")^1)+line.substring(bin,line.length());

                    }

                    line=**new** StringBuilder(line).reverse().toString();

                    System.out.println("Code Word: " + line + ".\n");

                }

                catch(IOException *i*){

                    break;

                }

            }

        }

        catch(IOException *i*){

        System.out.println(i);

        }

        System.out.println("Closing Connection.");

        try{

        socket.close();

        input.close();

        }

        catch(IOException *i*){

        System.out.println(i);

        }

    }

    @SuppressWarnings("unused")

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

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

    }

    public static *int* parity(String *line*,*int* *index*){

*int* p=0;

        for (*int* i=0;i<*line*.length();i++){

            String bin=**new** StringBuilder(Integer.toBinaryString(i+1)).reverse().toString();

            if (bin.length()>*index* && bin.charAt(*index*)=='1'){

                p=p^Integer.parseInt(*line*.charAt(i)+"");

            }

        }

        return p;

    }

}

### Client:

import java.net.*\**;

import java.io.*\**;

public class Client\_23BCE1145{

    private Socket socket=null;

    private DataInputStream input=null;

    private DataOutputStream output=null;

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

        try{

            System.out.println("Client has been started.");

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

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

            input=**new** DataInputStream(System.in);

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

        }

        catch(UnknownHostException *u*){

            System.out.println(u);

        return;

        }

        catch(IOException *u*){

            System.out.println(u);

        return;

        }

    }

    @SuppressWarnings("deprecation")

    public *void* Work(){

        String line="";

        while (!line.equalsIgnoreCase("END")){

            try{

                line=input.readLine();

                if (line.equalsIgnoreCase("END")){

                    break;

                }

*int* len=line.length();

*int* p=0;

                for (*int* i=1;i<len;i++){

                    if (Math.pow(2,i)>=(len+i+1)){

                        p=i;

                        break;

                    }

                }

                line=**new** StringBuilder(line).reverse().toString();

                System.out.println(len+p);

                for (*int* i=1;i<(len+p);i=i\*2){

                    line=line.substring(0,i-1)+'0'+line.substring(i-1,line.length());

                }

                for (*int* i=0;i<p;i++){

*int* pa=Even\_Parity(line,i);

                    System.out.println("P"+(*int*)Math.pow(2,i)+": "+pa);

                    line=line.substring(0,(*int*)Math.pow(2,i)-1)+pa+line.substring((*int*)Math.pow(2,i),line.length());

                }

                line=**new** StringBuilder(line).reverse().toString();

                System.out.println("Code Word: "+ line + ".\n");

                output.writeUTF(line);

            }

            catch(IOException *i*){

                break;

            }

        }

        try{

            System.out.println("Closing Connection.");

            socket.close();

            input.close();

            output.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();

    }

    public static *int* Even\_Parity(String *line*,*int* *index*){

*int* p=0;

        for (*int* i=0;i<*line*.length();i++){

            String bin=**new** StringBuilder(Integer.toBinaryString(i+1)).reverse().toString();

            if (bin.length()>*index* && bin.charAt(*index*)=='1'){

                p=p^Integer.parseInt(*line*.charAt(i)+"");

            }

        }

        return p;

    }

}

### Outputs:

A screenshot of a computer

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

AI-generated content may be incorrect.