**JAVA Assignment 2: MODULE 2**

**Name – CHINMAYA GARNAIK**

**Class - FYMCA(B)**

**PRN – 1132220942**

1.Write a simple of Java socket programming where client sends a text and server receives and prints it.

*Server.java:-*

import java.io.\*;

import java.net.\*;

public class q1\_server {

public static void main(String[] args) {

try {

ServerSocket listener = new ServerSocket(1880);

System.out.println("Server is ready");

Socket serverSocket = listener.accept();

InputStream input = serverSocket.getInputStream();

DataInputStream dis = new DataInputStream(input);

System.out.println(dis.readUTF());

listener.close();

} catch (IOException ie) {

ie.printStackTrace();

}

}

}

*Client.java:-*

import java.io.\*;

import java.net.\*;

import java.util.InputMismatchException;

import java.util.Scanner;

public class q1\_client {

public static void main(String[] args) {

try {

Socket clientSocket = new Socket("localhost", 1880);

OutputStream os = clientSocket.getOutputStream();

DataOutputStream dos = new DataOutputStream(os);

try (Scanner sc = new Scanner(System.in)) {

String msg = sc.nextLine();

dos.writeUTF(msg);

} catch (InputMismatchException ie) {

ie.printStackTrace();

}

clientSocket.close();

} catch (UnknownHostException ue) {

ue.printStackTrace();

} catch (IOException e) {

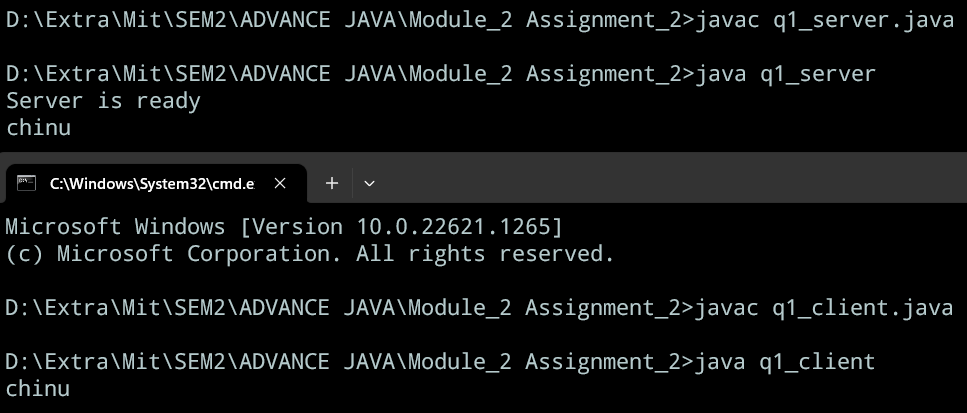
e.printStackTrace();

}

}

}

OUTPUT:-



2.Write a program to for client Server chat application

*Server.java:-*

import java.io.\*;

import java.net.\*;

public class q2\_server {

public static void main(String[] args) {

try {

ServerSocket serverSocket = new ServerSocket(8000);

System.out.println("Chat server started on port 8000");

Socket clientSocket = serverSocket.accept();

System.out.println("Client connected: " + clientSocket.getInetAddress().getHostAddress());

try (BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in))) {

String inputLine;

String outputLine;

// Start the conversation with the client

out.println("Welcome to the chat server!");

while ((inputLine = in.readLine()) != null) {

System.out.println("Client says: " + inputLine);

if (inputLine.equals("exit")) {

out.println("Goodbye!");

break;

} else {

System.out.print("Server says: ");

outputLine = stdIn.readLine();

out.println(outputLine);

}

}

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

*Client.java:-*

import java.io.\*;

import java.net.\*;

public class q2\_client {

public static void main(String[] args) {

try {

Socket socket = new Socket("localhost", 8000);

System.out.println("Connected to server");

try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in))) {

String inputLine;

while ((inputLine = in.readLine()) != null) {

System.out.println("Server says: " + inputLine);

if (inputLine.equals("Goodbye!")) {

break;

}

System.out.print("Client says: ");

String outputLine = stdIn.readLine();

out.println(outputLine);

}

} catch (UnknownHostException ue) {

ue.printStackTrace();

}

} catch (IOException e) {

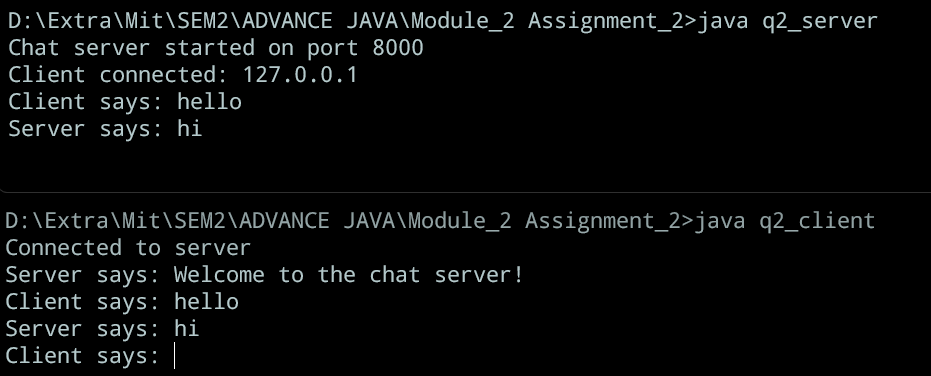
e.printStackTrace();

}

}

}

OUTPUT:-



3. Write a Server-side socket program to accept file name  
from client. Server will reverse the contents and send as a response.

*Server.java:-*

import java.io.\*;

import java.net.\*;

public class q3\_server {

public static void main(String[] args) {

try {

ServerSocket listener = new ServerSocket(1800);

System.out.println("Server is ready");

Socket serverSocket = listener.accept();

InputStream input = serverSocket.getInputStream();

DataInputStream dis = new DataInputStream(input);

OutputStream os = serverSocket.getOutputStream();

DataOutputStream dos = new DataOutputStream(os);

try {

String filename = dis.readUTF();

FileInputStream fi = new FileInputStream(filename);

BufferedReader br = new BufferedReader(new FileReader(filename));

String st;

while ((st = br.readLine()) != null) {

dos.writeUTF("File Content:" + st);

}

listener.close();

} catch (FileNotFoundException fe) {

fe.printStackTrace();

}

} catch (IOException ie) {

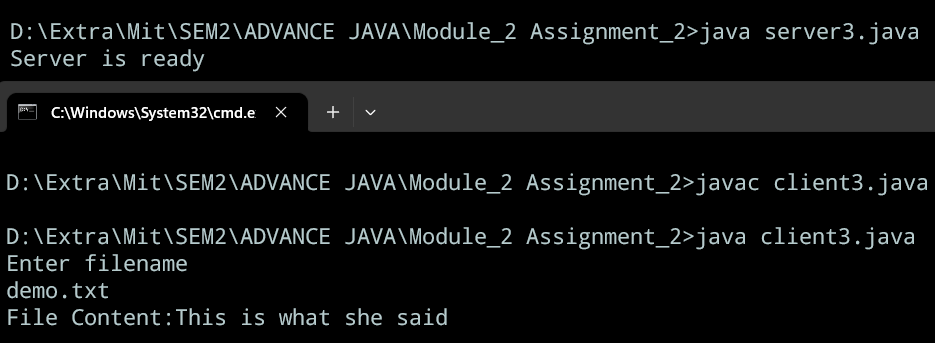
ie.printStackTrace();

}

}

}

OUTPUT:-



4. Write client-side  
socket program to accept filename from user and send to server. Display the  
response received from server.

*Client.java:-*

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class q4\_client {

public static void main(String abc[]) {

try {

Socket clientSocket = new Socket("localhost", 1990);

OutputStream os = clientSocket.getOutputStream();

DataOutputStream dos = new DataOutputStream(os);

InputStream is = clientSocket.getInputStream();

DataInputStream dis = new DataInputStream(is);

// for (int i = 0; i < 1; i++) {

Scanner scan = new Scanner(System.in);

System.out.println("Enter filename");

String msg = scan.nextLine();

dos.writeUTF(msg);

System.out.println(dis.readUTF());

} catch (UnknownHostException ue) {

ue.printStackTrace();

} catch (IOException ie) {

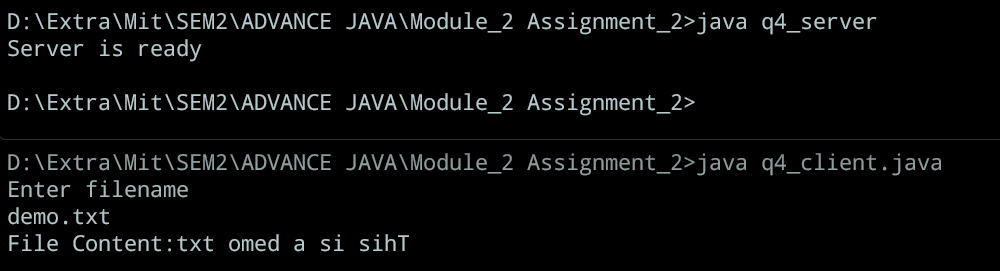
ie.printStackTrace();

}

}

}

OUTPUT:-



5. Write a program to accept string from client and replace  duplicate characters with X.  
Refer  
<https://www.javatpoint.com/socket-programming>

*Server.java:-*

import java.io.\*;

import java.net.\*;

import java.util.HashSet;

public class q5\_server {

public static String replaceDuplicateCharsWithX(String inputString) {

HashSet<Character> seen = new HashSet<>();

StringBuilder output = new StringBuilder();

for (char c : inputString.toCharArray()) {

if (seen.contains(c)) {

output.append('x');

} else {

output.append(c);

seen.add(c);

}

}

return output.toString();

}

public static void main(String[] args) {

try {

ServerSocket listener = new ServerSocket(1990);

System.out.println("Server is ready");

Socket serverSocket = listener.accept();

InputStream input = serverSocket.getInputStream();

DataInputStream dis = new DataInputStream(input);

OutputStream os = serverSocket.getOutputStream();

DataOutputStream dos = new DataOutputStream(os);

try {

String filename = dis.readUTF();

String outputString = replaceDuplicateCharsWithX(filename);

dos.writeUTF("OUTPUT:" + outputString);

listener.close();

} catch (FileNotFoundException fe) {

fe.printStackTrace();

}

} catch (IOException ie) {

ie.printStackTrace();

}

}

}

*Client.java:-*

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class q5\_client {

public static void main(String abc[]) {

try {

Socket clientSocket = new Socket("localhost", 1990);

OutputStream os = clientSocket.getOutputStream();

DataOutputStream dos = new DataOutputStream(os);

InputStream is = clientSocket.getInputStream();

DataInputStream dis = new DataInputStream(is);

Scanner scan = new Scanner(System.in);

System.out.println("Enter string");

String msg = scan.nextLine();

dos.writeUTF(msg);

System.out.println(dis.readUTF());

} catch (UnknownHostException ue) {

ue.printStackTrace();

} catch (IOException ie) {

ie.printStackTrace();

}

}

}

OUTPUT:-

