**Assignment No. 7**

* **HelloServer.java**

import java.io.\*;

import java.net.\*;

public class HelloServer {

public static void main(String[] args) {

try (ServerSocket serverSocket = new ServerSocket(12345)) {

System.out.println("Server is listening on port 12345...");

while (true) {

Socket socket = serverSocket.accept();

System.out.println("New client connected");

// Handle the client in a separate thread

new Thread(new ClientHandler(socket)).start();

}

}

catch (IOException e) {

System.out.println("Server exception: " + e.getMessage());

}

}

}

class ClientHandler implements Runnable {

private Socket socket;

public ClientHandler(Socket socket) {

this.socket = socket;

}

@Override

public void run() {

try (DataOutputStream out = new DataOutputStream(socket.getOutputStream())) {

out.writeUTF("Hello, client!");

}

catch (IOException e) {

System.out.println("Client handler exception: " + e.getMessage());

}

finally {

try {

socket.close();

}

catch (IOException e) {

System.out.println("Socket close exception: " + e.getMessage());

}

}

}

}

* **HelloClient.java**

import java.io.\*;

import java.net.\*;

public class HelloClient {

public static void main(String[] args) {

try (Socket socket = new Socket("localhost", 12345);

DataInputStream in = new DataInputStream(socket.getInputStream())) {

// Receive the greeting message from the server

String message = in.readUTF();

System.out.println("Message from server: " + message);

}

catch (IOException e) {

System.out.println("Client exception: " + e.getMessage());

}

}

}

**Output:**

PS E:\TE\CNS/PRACTICALS> javac HelloServer.java

PS E:\TE\CNS/PRACTICALS> java HelloServer

Server is listening on port 12345...

New client connected

PS E:\TE\CNS/PRACTICALS> javac HelloClient.java

PS E:\TE\CNS/PRACTICALS> java HelloClient

Message from server: Hello, client!

* **FileServer.java**

import java.io.\*;

import java.net.\*;

public class FileServer {

public static void main(String[] args) {

int port = 65432;

try (ServerSocket serverSocket = new ServerSocket(port)) {

System.out.println("Server is listening on port " + port);

while (true) {

try (Socket socket = serverSocket.accept()) {

System.out.println("New client connected");

// Receive the filename

DataInputStream dataInputStream = new DataInputStream(socket.getInputStream());

String fileName = dataInputStream.readUTF();

System.out.println("Receiving file: " + fileName);

// Create a file output stream to save the received file

FileOutputStream fos = new FileOutputStream(fileName);

BufferedOutputStream bos = new BufferedOutputStream(fos);

byte[] buffer = new byte[4096];

int bytesRead;

// Read from the socket and write to the file

while ((bytesRead = dataInputStream.read(buffer)) != -1) {

bos.write(buffer, 0, bytesRead);

}

bos.close();

System.out.println("File received successfully.");

}

catch (IOException e) {

System.out.println("Connection error: " + e.getMessage());

}

}

}

catch (IOException e) {

System.out.println("Server error: " + e.getMessage());

}

}

}

* **FileClient.java**

import java.io.\*;

import java.net.\*;

public class FileClient {

public static void main(String[] args) {

String host = "localhost"; // Change to the server's IP address if needed

int port = 65432;

String filePath = "C:/Users/91884/Desktop/5th Sem Books/CNS Lab/file.txt"; // Change to your file path

try (Socket socket = new Socket(host, port)) {

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

// Send the filename

DataOutputStream dataOutputStream = new DataOutputStream(socket.getOutputStream());

dataOutputStream.writeUTF(new File(filePath).getName());

// Send the file

FileInputStream fis = new FileInputStream(filePath);

BufferedInputStream bis = new BufferedInputStream(fis);

byte[] buffer = new byte[4096];

int bytesRead;

while ((bytesRead = bis.read(buffer)) != -1) {

dataOutputStream.write(buffer, 0, bytesRead);

}

bis.close();

System.out.println("File sent successfully.");

}

catch (IOException e) {

System.out.println("Client error: " + e.getMessage());

}

}

}

**Output:**

PS E:\TE\CNS/PRACTICALS> javac FileServer.java

PS E:\TE\CNS/PRACTICALS> java FileServer

Server is listening on port 65432

New client connected

Receiving file: file.txt

File received successfully.

PS E:\TE\CNS/PRACTICALS> javac FileClient.java

PS E:\TE\CNS/PRACTICALS> java FileClient

Connected to the server

Client error: C:\Users\91884\Desktop\5th Sem Books\CNS Lab\file.txt (The system cannot find the path specified)

* **CalculatorSever.java**

import java.io.\*;

import java.net.\*;

public class CalculatorServer {

public static void main(String[] args) {

int port = 65432;

try (ServerSocket serverSocket = new ServerSocket(port)) {

System.out.println("Calculator Server is listening on port " + port);

while (true) {

try (Socket socket = serverSocket.accept()) {

System.out.println("New client connected");

// Create input and output streams

BufferedReader input = new BufferedReader(new InputStreamReader(socket.getInputStream()));

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

String request;

while ((request = input.readLine()) != null) {

System.out.println("Received: " + request);

String response = calculate(request);

output.println(response);

}

}

catch (IOException e) {

System.out.println("Connection error: " + e.getMessage());

}

}

}

catch (IOException e) {

System.out.println("Server error: " + e.getMessage());

}

}

private static String calculate(String request) {

String[] tokens = request.split(" ");

if (tokens.length != 3) {

return "Invalid request. Use: <number1> <operator> <number2>";

}

try {

double num1 = Double.parseDouble(tokens[0]);

String operator = tokens[1];

double num2 = Double.parseDouble(tokens[2]);

switch (operator) {

case "+":

return String.valueOf(num1 + num2);

case "-":

return String.valueOf(num1 - num2);

case "\*":

return String.valueOf(num1 \* num2);

case "/":

return num2 != 0 ? String.valueOf(num1 / num2) : "Error: Division by zero";

default:

return "Invalid operator. Use +, -, \*, or /.";

}

}

catch (NumberFormatException e) {

return "Invalid number format.";

}

}

}

* **CalculatorClient.java**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class CalculatorClient {

public static void main(String[] args) {

String host = "localhost"; // Change to the server's IP address if needed

int port = 65432;

try (Socket socket = new Socket(host, port);

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

BufferedReader input = new BufferedReader(new InputStreamReader(socket.getInputStream()));

Scanner scanner = new Scanner(System.in)) {

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

String request;

while (true) {

System.out.print("Enter calculation (or 'exit' to quit): ");

request = scanner.nextLine();

if ("exit".equalsIgnoreCase(request)) {

break;

}

output.println(request);

String response = input.readLine();

System.out.println("Result: " + response);

}

}

catch (IOException e) {

System.out.println("Client error: " + e.getMessage());

}

}

}

**Output:**

PS E:\TE\CNS/PRACTICALS> javac CalculatorServer.java

PS E:\TE\CNS/PRACTICALS> java CalculatorServer

Calculator Server is listening on port 65432

New client connected

Received: 4+5

Received: 4 + 5

Received: 4 - 3

Received: 4 \* 5

Received: 20 / 5

PS E:\TE\CNS/PRACTICALS> javac CalculatorClient.java

PS E:\TE\CNS/PRACTICALS> java CalculatorClient

Connected to the calculator server

Enter calculation (or 'exit' to quit): 4+5

Result: Invalid request. Use: <number1> <operator> <number2>

Enter calculation (or 'exit' to quit): 4 + 5

Result: 9.0

Enter calculation (or 'exit' to quit): 4 - 3

Result: 1.0

Enter calculation (or 'exit' to quit): 4 \* 5

Result: 20.0

Enter calculation (or 'exit' to quit): 20 / 5

Result: 4.0

Enter calculation (or 'exit' to quit): exit