**Computer Networks**

# **Assignment-4**

# **Done By:** S. Vishwajith 23BCE1145

## Question-1:

## Code for Server:

import java.io.\*;

import java.net.\*;

public class Server{

    @SuppressWarnings("resource")

    public static void main(String[] args) throws IOException {

        ServerSocket serverSocket = new ServerSocket(5001);

        System.out.println("Server Listening on port 5001...");

        while (true) {

            Socket connectionSocket = serverSocket.accept();

            new Thread(() -> {

                try {

                    DataInputStream dataIn = new DataInputStream(connectionSocket.getInputStream());

                    DataOutputStream dataOut = new DataOutputStream(connectionSocket.getOutputStream());

                    String str = dataIn.readUTF();

                    String op = dataIn.readUTF();

                    switch(op){

                        case "Length":

                            dataOut.writeUTF(""+str.length());

                            break;

                        case "Upper":

                            dataOut.writeUTF(str.toUpperCase());

                            break;

                        case "Lower":

                            dataOut.writeUTF(str.toLowerCase());

                            break;

                        case "Append":

                            String str2 = "";

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

                                if(i%2==0){

                                    str2 += Character.toUpperCase(str.charAt(i));

                                }

                                else{

                                    str2 += str.charAt(i);

                                }

                            }

                            dataOut.writeUTF(str2);

                            break;

                        case "Reverse":

                            String str3 = "";

                            for(int i = str.length() - 1; i > -1; i--){

                                str3 += str.charAt(i);

                            }

                            dataOut.writeUTF(str3);

                            break;

                        default:

                            dataOut.writeUTF("Invalid Operation");

                            break;

                    }

                    connectionSocket.close();

                }

                catch(IOException e){

                    System.out.println(e);

                }

            }).start();

        }

    }

}

## Code for Client:

import java.io.\*;

import java.net.\*;

public class Client{

    private static Socket socket;

    private static DataInputStream In;

    private static DataOutputStream Out;

    public static void main(String [] args) throws IOException {

        String string = args[0];

        String Operation = args[1];

        socket = new Socket();

        socket.connect(new InetSocketAddress("127.0.0.1", 5001), 1000);

        System.out.println("Connection Successful!");

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

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

        Out.writeUTF(string);

        Out.writeUTF(Operation);

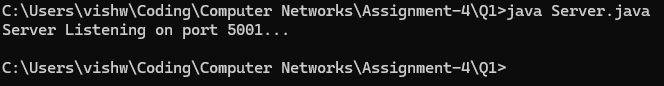
        String line = In.readUTF();

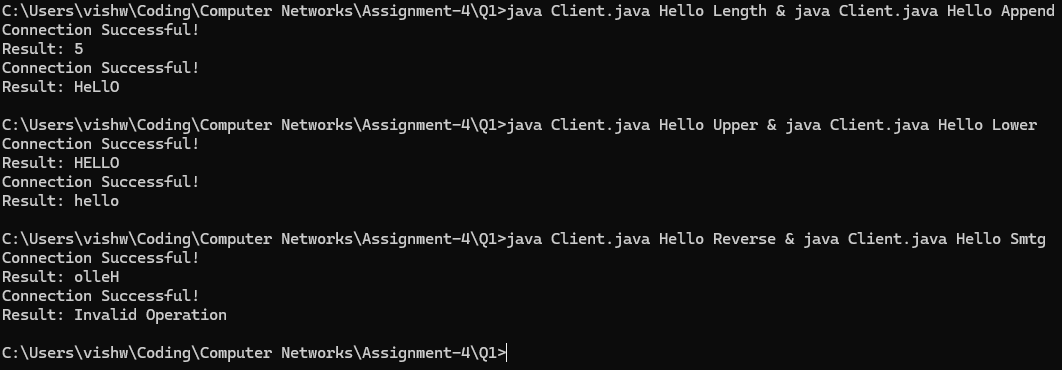
        System.out.println("Result: " + line);

    }

}

## Outputs:





## Question-2:

## Code for Server:

import java.io.\*;

import java.net.\*;

public class Server{

    private static ServerSocket serverSocket;

    private static Socket socket;

    private static DataInputStream In;

    private static DataOutputStream Out;

    public static void main(String [] args) throws IOException {

        serverSocket = new ServerSocket(5001);

        System.out.println("Server Listening on port 5001...");

        while (true) {

            socket = serverSocket.accept();

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

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

            new Thread(() -> {

                try {

                    String n = In.readUTF();

                    String base = In.readUTF();

                    switch(base){

                        case "Binary":

                            Out.writeUTF(Integer.toString(Integer.parseInt(n, 10), 2));

                            break;

                        case "Octal":

                            Out.writeUTF(Integer.toString(Integer.parseInt(n, 10), 8));

                            break;

                        case "Hexadecimal":

                            Out.writeUTF(Integer.toString(Integer.parseInt(n, 10), 16));

                            break;

                        default:

                            Out.writeUTF("Invalid Base");

                            break;

                    }

                    socket.close();

                }

                catch(IOException e){

                    System.out.println(e);

                }

            }).start();

        }

    }

}

## Code for Client:

import java.io.\*;

import java.net.\*;

public class Client{

    private static Socket socket;

    private static DataInputStream In;

    private static DataOutputStream Out;

    public static void main(String [] args) throws IOException {

        String n = args[0];

        String base = args[1];

        socket = new Socket();

        socket.connect(new InetSocketAddress("127.0.0.1", 5001), 1000);

        System.out.println("Connection Successful!");

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

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

        Out.writeUTF(n);

        Out.writeUTF(base);

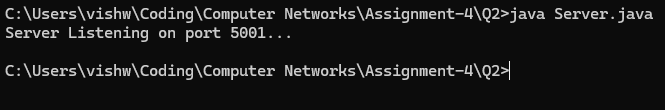
        String line = In.readUTF();

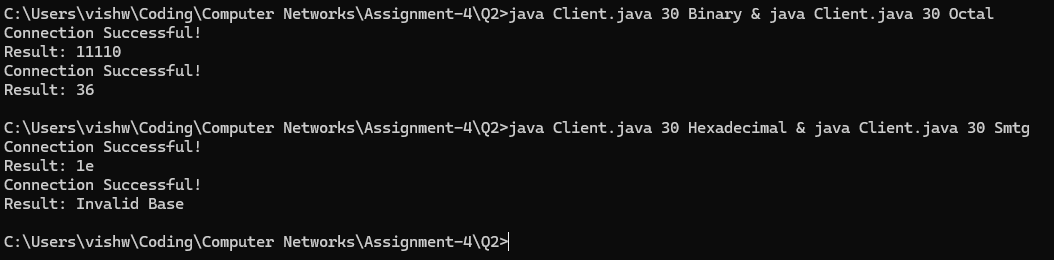
        System.out.println("Result: " + line);

    }

}

## Outputs:





## Question-3:

## Code for Server:

import java.io.\*;

import java.net.\*;

public class Server{

    @SuppressWarnings("resource")

    public static void main(String[] args) throws IOException {

        ServerSocket serverSocket = new ServerSocket(5001);

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

        while (true) {

            Socket clientSocket = serverSocket.accept();

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

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

        }

    }

}

class ClientHandler implements Runnable {

    private Socket clientSocket;

    public ClientHandler(Socket socket) {

        this.clientSocket = socket;

    }

    @Override

    public void run() {

        try {

            DataInputStream input = new DataInputStream(clientSocket.getInputStream());

            DataOutputStream output = new DataOutputStream(clientSocket.getOutputStream());

            double num1 = Double.parseDouble(input.readUTF());

            double num2 = Double.parseDouble(input.readUTF());

            String operator = input.readUTF();

            double result = 0;

            String line = "";

            switch (operator) {

                case "+":

                    result = num1 + num2;

                    break;

                case "-":

                    result = num1 - num2;

                    break;

                case "\*":

                    result = num1 \* num2;

                    break;

                case "/":

                    result = num1 / num2;

                    break;

                default:

                    line = "Invalid Operator";

            }

            if(line.equals("Invalid Operator")){

                output.writeUTF(line);

            }

            else{

                output.writeUTF("Result: " + result);

            }

        } catch (IOException e) {

            System.out.println(e);

        }

    }

}

## Code for Client:

import java.io.\*;

import java.net.\*;

public class Client{

  private static Socket socket;

  private static DataInputStream In;

  private static DataOutputStream Out;

  public static void main(String [] args) throws IOException {

    String num1 = args[0];

    String num2 = args[1];

    String operator = args[2];

    socket = new Socket();

    socket.connect(new InetSocketAddress("127.0.0.1", 5001), 1000);

    System.out.println("Connection Successful!");

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

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

    Out.writeUTF(num1);

    Out.writeUTF(num2);

    Out.writeUTF(operator);

    String line = In.readUTF();

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

  }

}

## Outputs:

