

Name: Tanmay Vig

Roll No.: 19BCS061

Class: 3rd year(5th sem)

Lab 5: WAP to implement Client Server to send and receive text messages from client and server.

Answer:

Source code:

Server: -

```
import java.io.*;

import java.net.*;

public class server{

    public static void main(String[] args){

        try{

            ServerSocket server = new ServerSocket(8080);

            Socket s = server.accept();

            DataInputStream in = new DataInputStream(s.getInputStream());

            DataOutputStream out = new DataOutputStream(s.getOutputStream());

            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            String str_recieve="",str_send="";

            while(!str_recieve.equals("stop")){

                str_recieve = in.readUTF();

                if(str_recieve.equals("stop")){

                    break;

                }

                System.out.println("Client: "+ str_recieve);

                str_send = br.readLine();

                out.writeUTF(str_send);

                out.flush();

            }

        }

    }

}
```

```

        System.out.println("Client: "+ str_recieve);

        out.close();

        in.close();

        server.close();

    }catch(Exception e){System.out.println("Connection Ended");}

    }

}

```

Client: -

```

import java.io.*;

import java.net.*;

public class client {

    public static void main(String[] args) {

        try{

            Socket s = new Socket("localhost",8080);

            DataOutputStream out = new DataOutputStream(s.getOutputStream());

            DataInputStream in = new DataInputStream(s.getInputStream());

            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            String str_send="",str_recieve="";

            while(!str_recieve.equals("stop")){

                str_send = br.readLine();

                out.writeUTF(str_send);

                out.flush();

                str_recieve = in.readUTF();

                System.out.println("Server: "+str_recieve);

            }

            out.close();

            s.close();

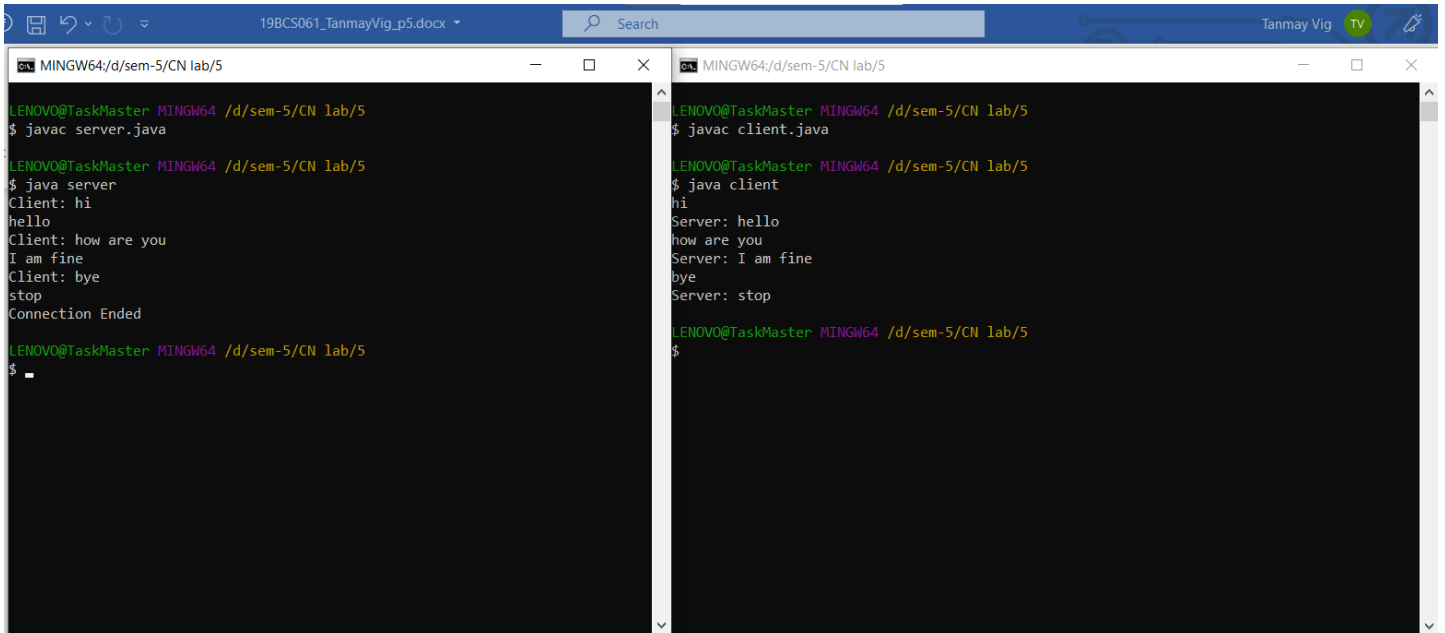
        }catch(Exception e){System.out.println("Connection Ended");}

    }

}

```

Output:



The image shows two side-by-side terminal windows from a Windows environment. The top bar of the application shows a document titled '19BCS061_TanmayVig_p5.docx' and a user profile 'Tanmay Vig' with a 'TV' icon. Both terminal windows have the title 'MINGW64/d/sem-5/CN lab/5'. The left terminal window shows the execution of a Java server program. It starts with 'javac server.java', then 'java server'. The output shows a client connecting and sending messages: 'hi', 'hello', 'Client: how are you', 'I am fine', 'Client: bye', 'stop', and 'Connection Ended'. The right terminal window shows the execution of a Java client program. It starts with 'javac client.java', then 'java client'. The output shows the client sending messages: 'hi', 'Server: hello', 'how are you', 'Server: I am fine', 'bye', and 'Server: stop'. Both windows end with a prompt '\$'.

```
LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$ javac server.java

LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$ java server
Client: hi
hello
Client: how are you
I am fine
Client: bye
stop
Connection Ended

LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$ -
```

```
LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$ javac client.java

LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$ java client
hi
Server: hello
how are you
Server: I am fine
bye
Server: stop

LENOVO@TaskMaster MINGW64 /d/sem-5/CN lab/5
$
```