Program

Server.java

```java
import java.net.*;
import java.io.*;

public class Server {
    // initialize socket and input stream
    private Socket socket = null;
    private ServerSocket server = null;
    private DataInputStream in = null;

    // constructor with port
    public Server(int port) {
        // starts server and waits for a connection
        try {
            server = new ServerSocket(port);
            System.out.println("Server socket created");

            socket = server.accept();

            // takes input from the client socket
            in = new DataInputStream(new
BufferedInputStream(socket.getInputStream()));

            String line = "";

            // reads message from client until "Over" is sent
            while (!line.equals("Over")) {
                try {
                    line = in.readUTF();
                    System.out.println("From Client: " + line);

                } catch (IOException i) {
                }
            }

            // close connection
            socket.close();
            in.close();
        } catch (IOException i) {
        }
    }

    public static void main(String args[]) {
        Server server = new Server(5000);
    }
}
```

Client.java

```java
import java.net.*;
import java.io.*;

public class Client {
    // initialize socket and input output streams
    private Socket socket = null;
    private DataInputStream input = null;
    private DataOutputStream out = null;

    // constructor to put ip address and port
    public Client(String address, int port) {
        // establish a connection
        try {
            socket = new Socket(address, port);
            System.out.println("Message to Server: ");

            // takes input from terminal
            input = new DataInputStream(System.in);

            // sends output to the socket
            out = new DataOutputStream(socket.getOutputStream());
        } catch (UnknownHostException u) {
            System.out.println(u);
        } catch (IOException i) {
            System.out.println(i);
        }

        // string to read message from input
        String line = "";

        // keep reading until "Over" is input
        while (!line.equals("Exit")) {
            try {
                line = input.readLine();
                out.writeUTF(line);
            } catch (IOException i) {
                System.out.println(i);
            }
        }

        // close the connection
        try {
            input.close();
            out.close();
            socket.close();
        } catch (IOException i) {
            System.out.println(i);
        }
    }

    public static void main(String args[]) {
```
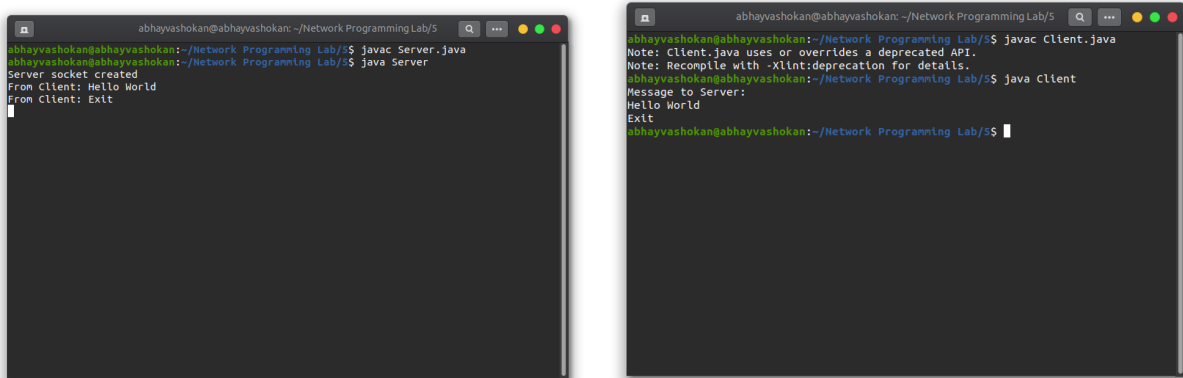
```
        Client client = new Client("127.0.0.1", 5000);
    }
}
```

Output

Screenshots

Output

Server

```
Server socket created
From Client: Hello World
From Client: Exit
```

Client

```
Message to Server:
Hello World
Exit
```

ReadMe

1. Open first terminal

    1. `javac Server.java`
    2. `java Server`

2. Open second terminal

    1. `javac Client.java`
    2. `java Client`

3. Communicate between Client and Server using the terminal.

4. To exit type: `Exit` in Client.