Experiment – 3.1

Student Name: Deepak Saini **UID:** 20BCS4066
Branch: 20BCC1 **Section/Group:** A
Semester: 5th **Date of Performance:** 11/10/2022
Subject Name: Computer Networks Lab **Subject Code:** 20CSP-342

1.Aim/Overview of the Practical

Using Socket programming implement the Connection oriented service using standard Ports inany programming language (Java/Python etc).

2. Task to be Done

Using Socket programming implement the Connection oriented service using standard Ports inany programming language (Java/Python etc).

3. Application

Requirements:
PC, JDK, CMD/Terminal or any JAVA IDE like IntelliJ.

4. Theory:

Java Networking: Java Networking is a concept of connecting two or more computingdevices together so that we can share resources.
Java socket programming provides facility to share data between different computingdevices.

The java.net package supports two protocols

TCP: Transmission Control Protocol provides reliable communication between the senderand receiver. TCP is used along with the Internet Protocol referred as TCP/IP.
UDP: User Datagram Protocol provides a connection-less protocol service by allowing packet of data to be transferred along two or more nodes.

Socket: A socket is an endpoint between two-way communications. Visit next page for Javasocket programming.

java.net package: The java.net package provides many classes to deal with networking applications in Java.

5. Steps for the practical/ Result/ Output:

1. Open any text editor or any JAVA Supported IDE.
2. Add the code given below and save file as MyServer and MyClient respectively with .java extension.
3. Run the code on to the IDE or CMD, first run the server program then client.
4. Send message from client side and do the same from the server side.
5. If the connection is successful then we should be able to see the messages from bothsides.

Code:

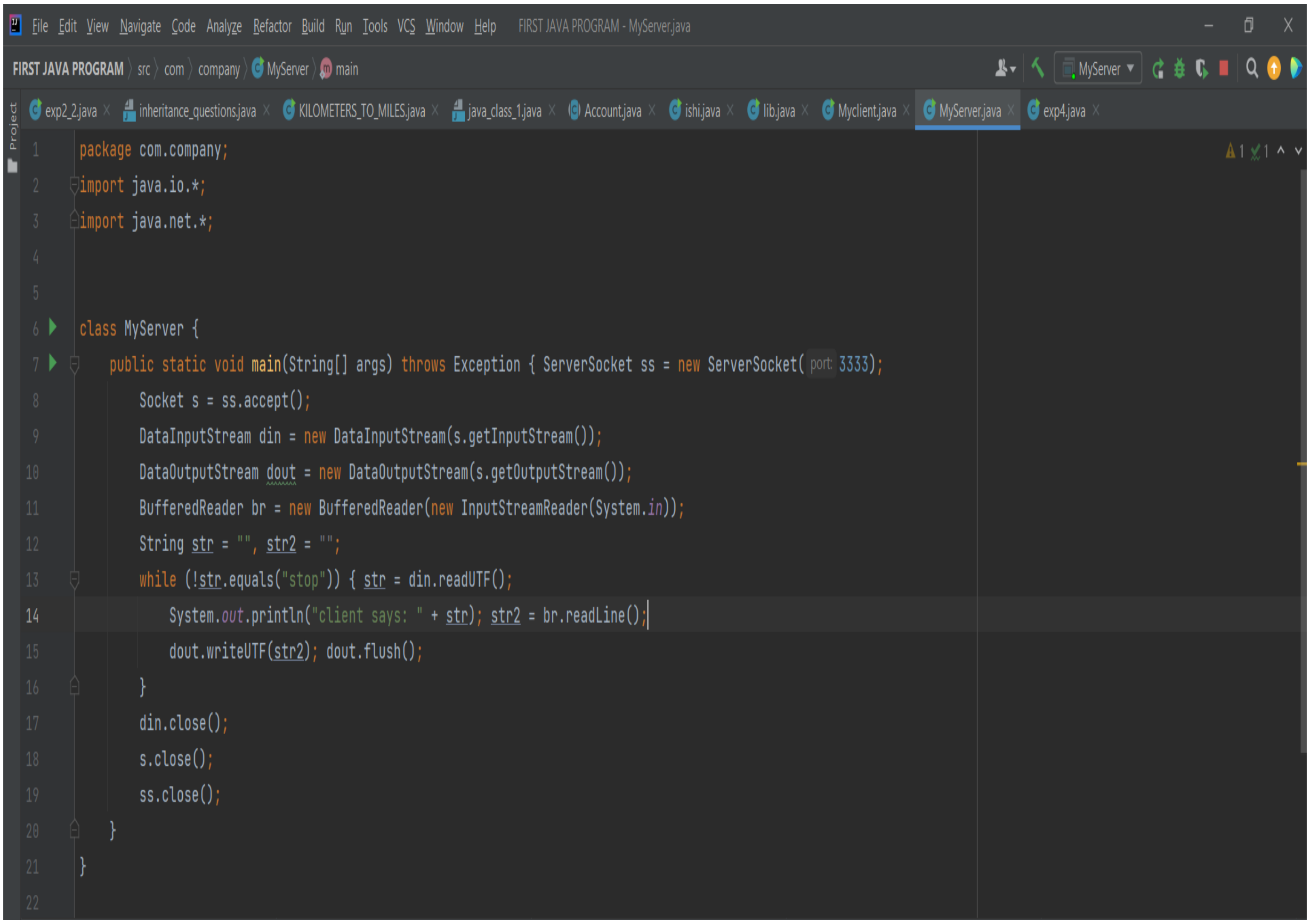
a) MyServer.java

```
package com.company;

import java.io.*;

import java.net.*;

class MyServer {

    public static void main(String[] args) throws Exception { ServerSocket ss = new
ServerSocket(3333);

        Socket s = ss.accept();

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

        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String str = "", str2 = "";

        while (!str.equals("stop")) { str = din.readUTF();

            System.out.println("client says: " + str); str2 = br.readLine();

            dout.writeUTF(str2); dout.flush();

        }

        din.close();

        s.close();

        ss.close();

    }

}
```

b) MyClient.java :

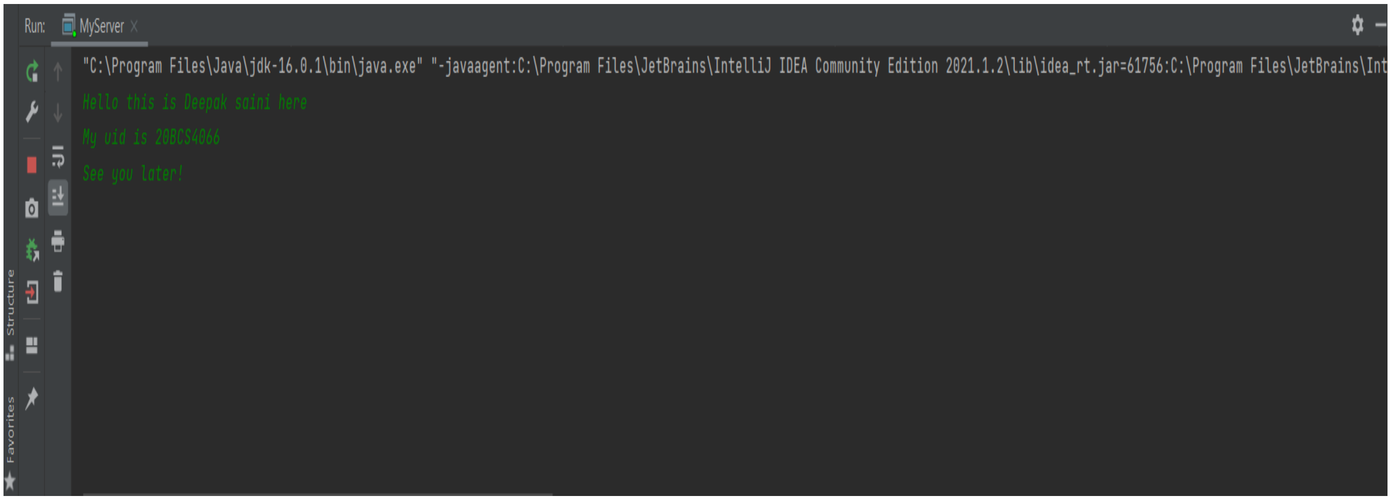
```
package com.company;

import java.net.*;

import java.io.*;



class MyClient {

    public static void main(String[] args) throws Exception { Socket s = new
Socket("localhost", 3333);

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

        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String str = "", str2 = ""; while (!str.equals("stop")) {

            str = br.readLine(); dout.writeUTF(str); dout.flush();

            str2 = din.readUTF(); System.out.println("Server says: " + str2);

        }

        dout.close();

        s.close();

    }

}
```

Output:

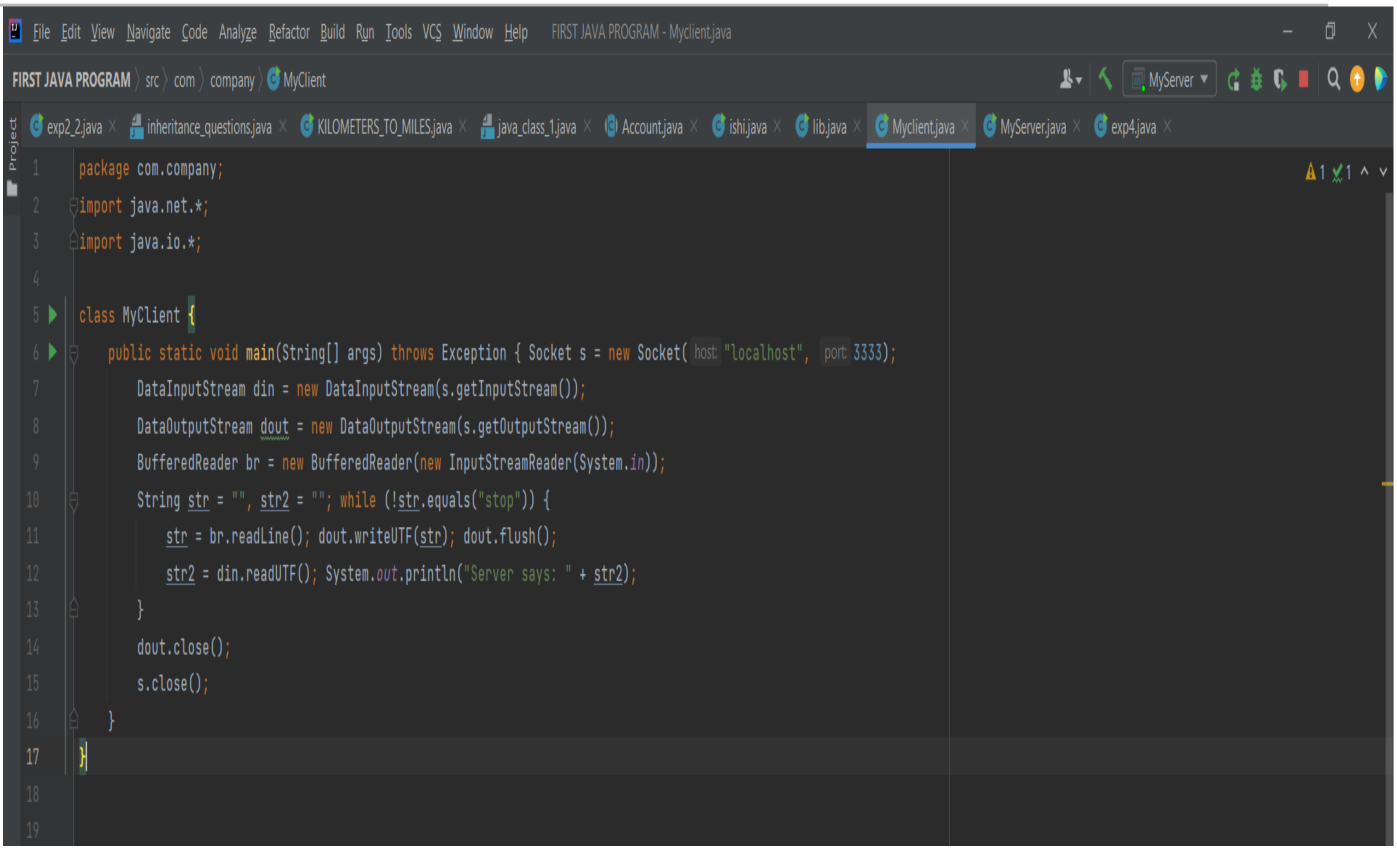


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help FIRST JAVA PROGRAM - MyServer.java
FIRST JAVA PROGRAM 〉 src 〉 com 〉 company 〉 MyServer 〉 main
MyServer
exp2_2.java
inheritance_questions.java
KILOMETERS_TO_MILES.java
java_class_1.java
Account.java
ishi.java
lib.java
Myclient.java
MyServer.java
exp4.java

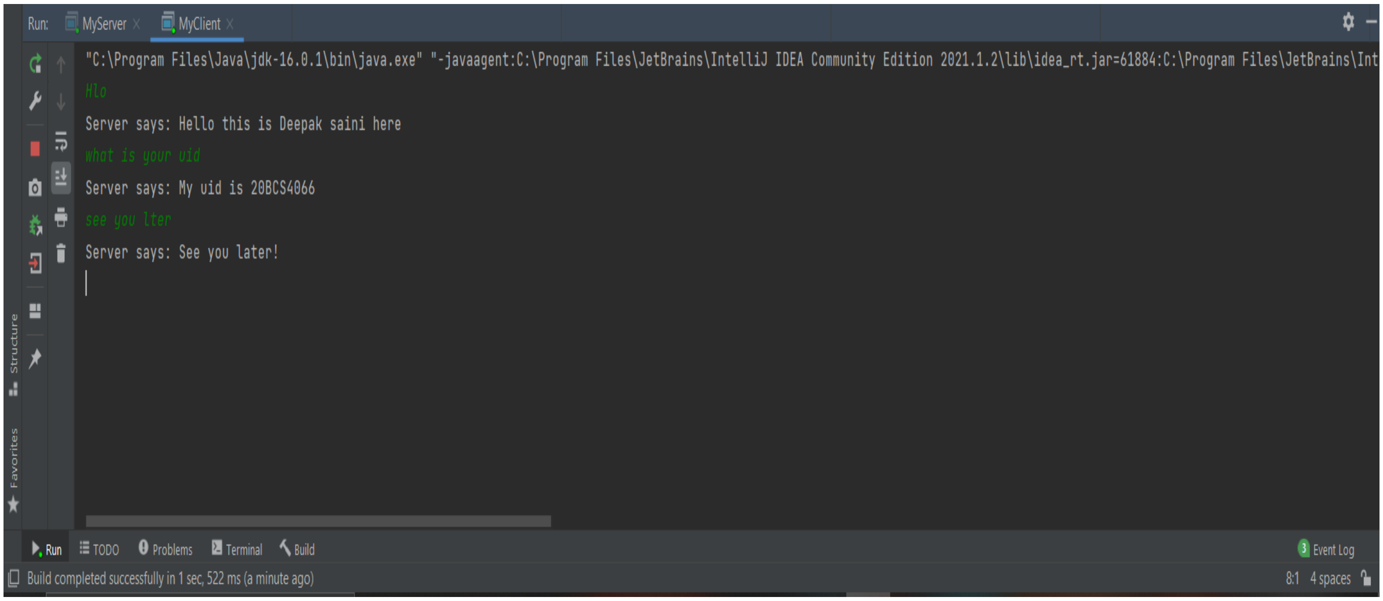
```java
package com.company;
import java.io.*;
import java.net.*;


class MyServer {
    public static void main(String[] args) throws Exception { ServerSocket ss = new ServerSocket( port: 3333);
        Socket s = ss.accept();
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String str = "", str2 = "";
        while (!str.equals("stop")) { str = din.readUTF();
            System.out.println("client says: " + str); str2 = br.readLine();
            dout.writeUTF(str2); dout.flush();
        }
        din.close();
        s.close();
        ss.close();
    }
}
```



```
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.2\lib\idea_rt.jar=61756:C:\Program Files\JetBrains\Int
Hello this is Deepak saini here
My uid is 20BCS4066
See you later!
```


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help FIRST JAVA PROGRAM - Myclient.java
FIRST JAVA PROGRAM 〉 src 〉 com 〉 company 〉 MyClient
MyServer
exp2_2.java
inheritance_questions.java
KILOMETERS_TO_MILES.java
java_class_1.java
Account.java
ishi.java
lib.java
Myclient.java
MyServer.java
exp4.java
Project
▲1 ✓1

```java
package com.company;
import java.net.*;
import java.io.*;

class MyClient {
    public static void main(String[] args) throws Exception { Socket s = new Socket( host: "localhost",  port: 3333);
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String str = "", str2 = ""; while (!str.equals("stop")) {
            str = br.readLine(); dout.writeUTF(str); dout.flush();
            str2 = din.readUTF(); System.out.println("Server says: " + str2);
        }
        dout.close();
        s.close();
    }
}
```



```
"C:\Program Files\Java\jdk-16.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.2\lib\idea_rt.jar=61884:C:\Program Files\JetBrains\Int
Hlo
Server says: Hello this is Deepak saini here
what is your uid
Server says: My uid is 20BCS4066
see you lter
Server says: See you later!
```

Build completed successfully in 1 sec, 522 ms (a minute ago)

Learning Outcomes:

1. Leant how to establish connection using java.
2. Learnt the creation of connection ports.
3. Learned about different networking libraries of JAVA.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			