**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 in any programming language (Java/Python etc).

# 2. Task to be Done

Using Socket programming implement the Connection oriented service using standard Ports in any 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 computing devices together so that we can share resources.

Java socket programming provides facility to share data between different computing devices.

# The java.net package supports two protocols

**TCP:** Transmission Control Protocol provides reliable communication between the sender and 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 Java socket 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.

* 1. Run the code on to the IDE or CMD, first run the server program then client.
  2. Send message from client side and do the same from the server side.
  3. If the connection is successful then we should be able to see the messages from both sides.

# Code:

# 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:

# 

# 

# 

# 

# 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. |  |  |  |
|  |  |  |  |