EXPERIMENT NO. 8

SEMESTER: V **DATE OF PERFORMANCE: 25th September 2024**

SUBJECT: CN LAB **DATE OF SUBMISSION: 05th October 2024**

NAME OF THE STUDENT: DAVID JAMES **ROLL NO: 22**

AIM	Socket Programming using UDP
LEARNING OBJECTIVE	Students will be able to able to develop a chat application using UDP protocols.
LEARNING OUTCOME	The students will be able to write client server chat application program using TCP.
COURSE OUTCOME	CSL502.4: Illustrate socket programming for TCP/UDP connections fordemonstrating networking concepts
PROGRAM OUTCOME	PO1,PO2,PO3,PO4,PO5,PO9,PO10,PSO1,PSO2,PSO3
BLOOM'S TAXONOMY LEVEL	Analyze
THEORY	**Java Socket Programming** • Java Socket programming is used for communication between the applications running on different JRE. • Java Socket programming can be connection-oriented or connection-less. • Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming. The client in socket programming must know two information: a. IP Address of Server, and b. Port number. Here, we are going to make one-way client and server communication. In this application, client sends a message to the server, server reads the message and printsit. Here, two classes are being used: Socket and ServerSocket. The Socket class is used to communicate client and server. Through this class, we can read and write message. The ServerSocket class is used at server-side. The accept() method of ServerSocket class blocks the console until the client is connected. After the successful connection of client, it returns the instance of Socket at server-side. **#Socket class** A socket is simply an endpoint for communications between the machines. The Socketclass can be used to create a socket. **#ServerSocket class** The ServerSocket class can be used to create a server socket. This object is used toestablish communication with the clients.

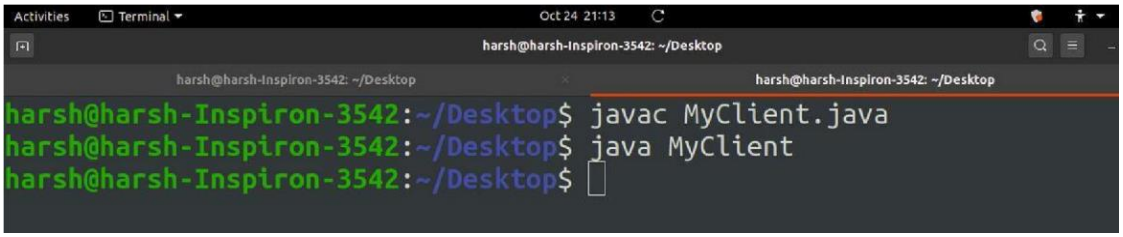
Creating Server:

To create the server application, we need to create the instance of ServerSocket class.Here, we are using 6666 port number for the communication between the client and server. You may also choose any other port number. The accept() method waits for theclient. If clients connects with the given port number, it returns an instance of Socket.ServerSocket ss=**new** ServerSocket(6666);
Socket s=ss.accept();//establishes connection and waits for the client

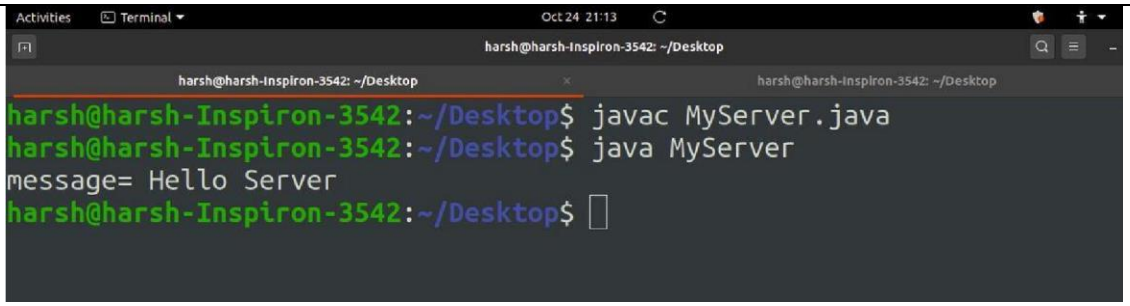
Creating Client:
To create the client application, we need to create the instance of Socket class. Here, we need to pass the IP address or hostname of the Server and a port number. Here, weare using "localhost" because our server is running on same system.
Socket s=**new** Socket("localhost",6666);

Output:
To execute this program open two command prompts and execute each program ateach command prompt as displayed in the below figures. First run Myserver.java file in terminal/cmd,

Running MyServer.java
Then in new terminal/cmd run MyClient.java file,



Running MyClient.java
As soon as you run MyClient program a message is sent to server anddisplayed in MyServer Terminal/CMD as shown below,



Message displayed in MyServer after running MyClient

LAB EXERCISE

Write a program to build chat application using SOCKET programming (UDP).

Client Side Program: udpBaseClient_2.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class udpBaseClient_2 {
    public static void main(String args[]) throws IOException {
        Scanner sc = new Scanner(System.in);

        // Step 1: Create the socket object for carrying the data.
        DatagramSocket ds = new DatagramSocket();
        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;

        // Loop while user has not entered "bye"
        while (true) {
            String inp = sc.nextLine();

            // Convert the String input into the byte array.
            buf = inp.getBytes();

            // Step 2: Create the DatagramPacket for sending the data.
            DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, 1234);

            // Step 3: Invoke the send call to actually send the data.
            ds.send(DpSend);

            // Break the loop if user enters "bye"
            if (inp.equals("bye")) {
                break;
```

```
            }
        }
        sc.close();
        ds.close();
    }
}
```

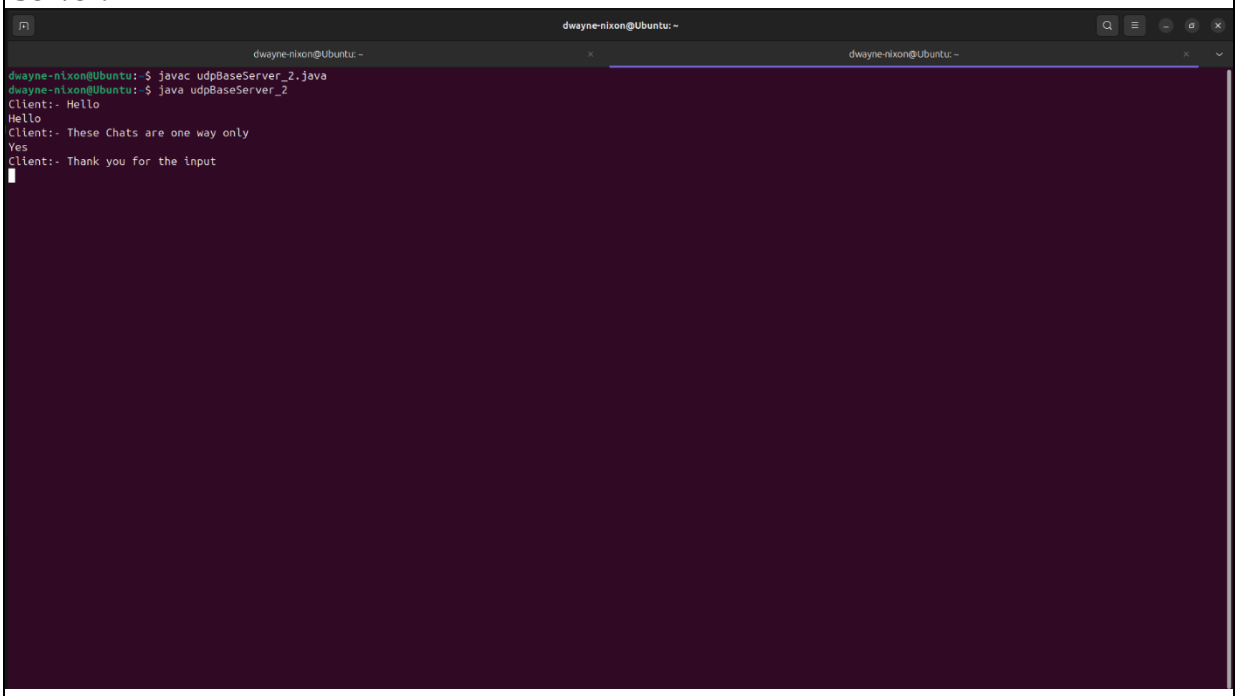
Server Side Program: udpBaseServer_2.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

public class udpBaseServer_2 {
    public static void main(String[] args) throws IOException {
        // Step 1: Create a socket to listen at port 1234
        DatagramSocket ds = new DatagramSocket(1234);
        byte[] receive = new byte[65535];

        DatagramPacket DpReceive = null;

        while (true) {
            // Step 2: Create a DatagramPacket to receive the data.
            DpReceive = new DatagramPacket(receive, receive.length);

            // Step 3: Receive the data in byte buffer.
            ds.receive(DpReceive);

            System.out.println("Client:- " + data(receive));

            // Exit the server if the client sends "bye"
            if (data(receive).toString().trim().equals("bye")) {
                System.out.println("Client sent bye... EXITING");
                break;
            }

            // Clear the buffer after every message.
            receive = new byte[65535];
        }
        ds.close();
    }

    // A utility method to convert the byte array data into a string representation.
    public static StringBuilder data(byte[] a) {
        if (a == null) return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
```

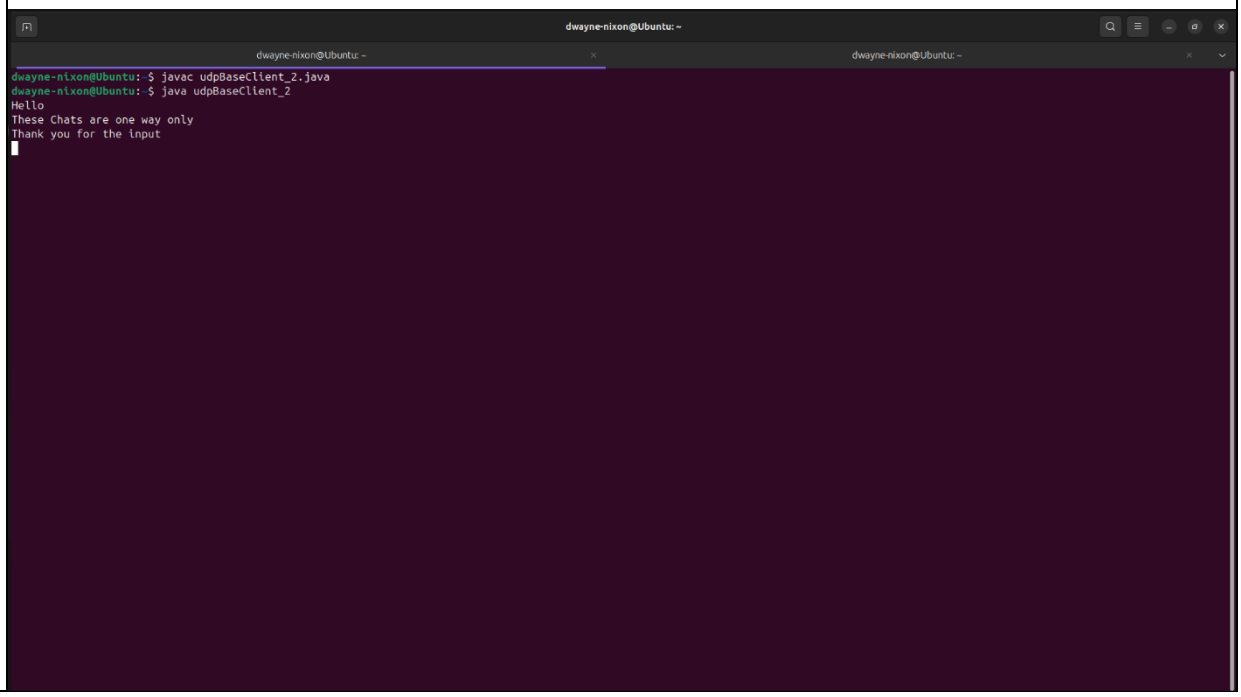
```
        while (a[i] != 0) {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}
```

Output:

Server:



Client:



REFERENCES	• B.A. Forouzan, "Data Communications and Networking", TMH, FourthEdition. • https://www.tutorialspoint.com/unix_sockets/what_is_socket.htm • https://www.geeksforgeeks.org/working-udp-datagramsockets-java/ https://www.youtube.com/watch?v=UaM1JmQliTs