**Experiment 05: Socket Programming**

1. **1 way communication: Client to Server Only**

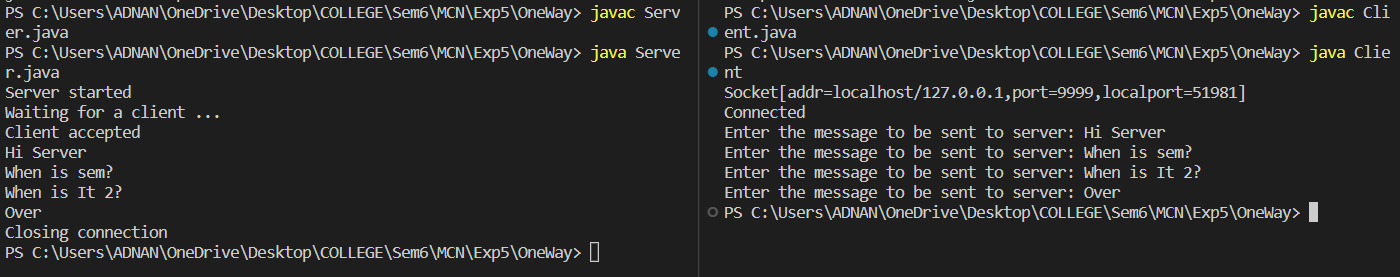
**Client Code:**

// A Java program for a Client  
import java.net.\*;  
import java.io.\*;  
import java.util.\*;  
  
public class Client {  
    // initialize socket and input output streams  
    private Socket socket = null;  
    private DataInputStream input = null;  
    private DataOutputStream out = null;  
    private Scanner sc = 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(socket);  
            System.out.println("Connected");  
  
            // takes input from terminal  
            input = new DataInputStream(System.in);  
  
            // sends output to the socket  
            out = new DataOutputStream(socket.getOutputStream());  
  
            sc = new Scanner(System.in);  
        } 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.equalsIgnoreCase("Over")) {  
            try {  
                System.out.print("Enter the message to be sent to server: ");  
                line = sc.nextLine();  
                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[]) {  
  
        new Client("localhost", 9999);  
    }  
}

**Server Code:**

// A Java program for a Server  
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 started");  
  
            System.out.println("Waiting for a client ...");  
  
            socket = server.accept();  
            System.out.println("Client accepted");  
  
// 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.equalsIgnoreCase("Over"))  
            {  
                try  
                {  
                    line = in.readUTF();  
                    System.out.println(line);  
  
                }  
                catch(IOException i)  
                {  
                    System.out.println(i);  
                    break;  
                }  
            }  
            System.out.println("Closing connection");  
  
// close connection  
            socket.close();  
            in.close();  
        }  
        catch(IOException i)  
        {  
            System.out.println("Client Left");  
        }  
    }  
  
    public static void main(String args[])  
    {  
        Server server = new Server(9999);  
    }  
}

**Output: LeftSide: Server, RightSide: Client**



1. **2 way communication: Client to Server And Back**

**Client Code:**

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class Client

{

// initialize socket and input output streams

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream out = null;

private Scanner sc = 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("Connected to server");

// read what server sends

input = new DataInputStream(socket.getInputStream());

// sends message to the server

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

// Take i/p from terminal

sc = new Scanner(System.in);

}

catch (Exception e)

{

System.out.println("AN\_\_\_\_ E R R O R\_\_\_\_OCCURRED: " + e);

}

// string to read message from input

String ClientMessage = "", ServerMessage = "";

sc = new Scanner(System.in);

// keep reading until "Over" is input

while (!ClientMessage.equals("Over") && !ServerMessage.equals("Over"))

{

try

{

System.out.println("Enter message to be sent to server: ");

ClientMessage = sc.nextLine(); // Take line to be sent to server

if (ClientMessage.equals("Over"))

{

System.out.println("Client wants to leave!");

}

out.writeUTF(ClientMessage); // Send to server

ServerMessage = input.readUTF(); // Receive message from server

System.out.println("Server replied: ---> " + ServerMessage); // display it

}

catch (Exception e)

{

// System.out.println("AN\_\_\_\_ E R R O R\_\_\_\_OCCURRED: " + e);

}

}

// close the connection

try

{

input.close();

out.close();

socket.close();

sc.close();

}

catch (IOException i)

{

System.out.println(i);

}

}

public static void main(String args[])

{ new Client("localhost", 6666); }

}

**Server Code:**

import java.net.\*;

import java.io.\*;

import java.util.\*;

public class Server {

// initialize socket and input stream

private Socket socket = null;

private ServerSocket server = null;

private DataInputStream in = null; // To get ClientMessage

private DataOutputStream out = null; // To write ServerMessage

private Scanner sc = null; // To take server message as i/p

// constructor with port

public Server(int port) {

// starts server and waits for a connection

try {

server = new ServerSocket(port);

System.out.println("Server started");

System.out.println("Waiting for a client ...");

socket = server.accept();

System.out.println("Client accepted");

// Help us reading ClientMessage

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

// Help up sending ServerMessage to Client

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

String ClientMessage = "", ServerMessage = "";

// reads message from client until "Over" is sent or server wants to over the

// communication

while (!ClientMessage..equalsIgnoreCase("Over") && !ServerMessage.equalsIgnoreCase("Over")) {

try {

ClientMessage = in.readUTF(); // Read what client sent

if (ClientMessage.equalsIgnoreCase("Over")) {

System.out.println("Client left!");

continue;

}

// Display what client sent

System.out.println("Client Says : " + ClientMessage);

// Take i/p the server message

System.out.println("Enter message to be sent to client: ");

sc = new Scanner(System.in);

ServerMessage = sc.nextLine();

// Send server message to the client via socket

out.writeUTF(ServerMessage);

} catch (IOException e) {

System.out.println("Client left");

}

}

System.out.println("Closing connection");

// close connection

socket.close();

in.close();

sc.close();

} catch (IOException i) {

System.out.println("Client left");

}

}

public static void main(String args[]) {

try{

new Server(6666);

}catch(Exception e){

System.out.println("Client has left");

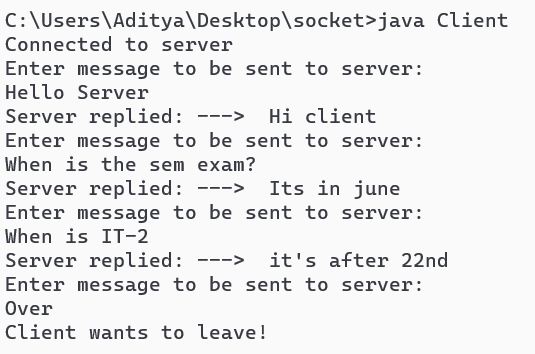
}

}

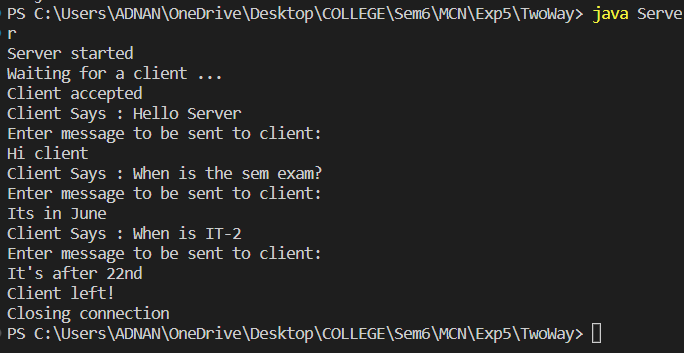
}

**Output:**

**Client Side:**



**Server Side:**

****