**Practical - 6**

**Aim: Implement a UDP based client server program in which client send one string to server, server will return the reverse of received string to client.**

**Code:**

***Client.java***

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class Client {

public static void main(String args[]) throws Exception {

// Create a socket to connect to the server

DatagramSocket clientSocket = new DatagramSocket();

// Get the IP address of the server

InetAddress IPAddress = InetAddress.getByName("localhost");

// Create a buffer to hold the message

byte[] sendData = new byte[1024];

// Create a buffer to hold the incoming message

byte[] receiveData = new byte[1024];

// Create a scanner to read the message from the user

Scanner inFromUser = new Scanner(System.in);

// Read the message from the user

String sentence = inFromUser.nextLine();

// Convert the message to bytes

sendData = sentence.getBytes();

// Create a datagram packet to send the message

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

// Send the message to the server

clientSocket.send(sendPacket);

// Create a datagram packet to receive the message

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

// Receive the message from the server

clientSocket.receive(receivePacket);

// Convert the message to a string

String modifiedSentence = new String(receivePacket.getData());

// Print the message

System.out.println("Server sent: " + modifiedSentence);

// Close the socket

clientSocket.close();

}

}

***UDPServer.java***

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class UDPServer {

public static void main(String args[]) throws Exception {

// Create a socket to listen for connections

DatagramSocket serverSocket = new DatagramSocket(9876);

// Create a buffer to hold the incoming message

byte[] receiveData = new byte[1024];

// Create a buffer to hold the message to send

byte[] sendData = new byte[1024];

// Create a datagram packet to receive the message

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

System.out.println("Waiting for client on port " + serverSocket.getLocalPort() + "...");

// Receive the message from the client

serverSocket.receive(receivePacket);

System.out.println("Client connected from " + receivePacket.getAddress());

// Convert the message to a string

String sentence = new String(receivePacket.getData());

System.out.println("Client sent: " + sentence);

// Get the IP address of the client

InetAddress IPAddress = receivePacket.getAddress();

// Get the port number of the client

int port = receivePacket.getPort();

// Reverse the message

String reversedSentence = new StringBuffer(sentence).reverse().toString();

// Convert the message to bytes

sendData = reversedSentence.getBytes();

// Create a datagram packet to send the message

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);

// Send the message to the client

System.out.println("Sending message back to client: " + reversedSentence);

serverSocket.send(sendPacket);

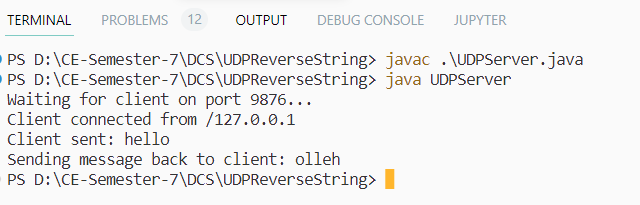
// Close the socket

serverSocket.close();

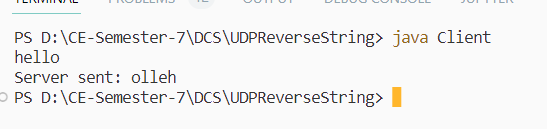
}

}

**UDPServer**

******

**Client**

******