**Practical 1.6**

**Aim:**

Write a java program using java.net library. Client Write a client program to send any string from its standard input to the server program. The server program reads the string, finds number of characters and digits and sends it back to client program. Use connection-oriented and connection-less communication.

**PROGRAM CODE:**

1. **Connection-Oriented Communication:**

**TCP Client:**

**import** java**.**io**.\*;**

**import** java**.**net**.\*;**

public class CountClient **{**

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

String serverHostname **=** **new** String **(**"127.0.0.1"**);**

**if** **(**args**.**length **>** 0**)**

serverHostname **=** args**[**0**];**

System**.**out**.**println **(**"Attemping to connect to host " **+**

serverHostname **+** " on port 10007."**);**

Socket echoSocket **=** **null;**

PrintWriter out **=** **null;**

BufferedReader in **=** **null;**

**try** **{**

// echoSocket = new Socket("taranis", 7);

echoSocket **=** **new** Socket**(**serverHostname**,** 10007**);**

out **=** **new** PrintWriter**(**echoSocket**.**getOutputStream**(),** **true);**

in **=** **new** BufferedReader**(new** InputStreamReader**(**

echoSocket**.**getInputStream**()));**

**}**

**catch** **(**UnknownHostException e**)** **{**

System**.**err**.**println**(**"Don't know about host: " **+** serverHostname**);**

System**.**exit**(**1**);**

**}**

**catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"Couldn't get I/O for "

**+** "the connection to: " **+** serverHostname**);**

e**.**printStackTrace**();**

System**.**exit**(**1**);**

**}**

BufferedReader stdIn **=** **new** BufferedReader**(new** InputStreamReader**(**System**.**in**));**

String userInput**;**

System**.**out**.**print **(**"input: "**);**

userInput **=** stdIn**.**readLine**();**

out**.**println**(**userInput**);**

System**.**out**.**println**(**"output: " **+** in**.**readLine**());**

out**.**close**();**

in**.**close**();**

stdIn**.**close**();**

echoSocket**.**close**();**

**}**

**}**

**TCP Server:**

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

public class CountServer **{**

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

ServerSocket serverSocket **=** **null;**

**try** **{**

serverSocket **=** **new** ServerSocket**(**10007**);**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Could not listen on port: 10007."**);**

System**.**exit**(**1**);**

**}**

Socket clientSocket **=** **null;**

System**.**out**.**println **(**"Waiting for connection....."**);**

**try** **{**

clientSocket **=** serverSocket**.**accept**();**

**}**

**catch** **(**IOException e**)**

**{**

System**.**err**.**println**(**"Accept failed."**);**

System**.**exit**(**1**);**

**}**

System**.**out**.**println **(**"Connection successful"**);**

System**.**out**.**println **(**"Waiting for input....."**);**

PrintWriter out **=** **new** PrintWriter**(**clientSocket**.**getOutputStream**(),** **true);**

BufferedReader in **=** **new** BufferedReader**(new** InputStreamReader**(**clientSocket**.**getInputStream**()));**

String inputLine**;**

**while** **((**inputLine **=** in**.**readLine**())** **!=** **null){**

System**.**out**.**println **(**"Server: " **+** inputLine**);**

int i**=**0**,**d**=**0**,**c**=**0**;**

**while(**i**<=**inputLine**.**length**()-**1**){**

**if** **(**Character**.**isDigit**(**inputLine**.**charAt**(**i**)))** **{**

d**++;**

**}**

**else** **if** **(**Character**.**isLetter**(**inputLine**.**charAt**(**i**)))** **{**

c**++;**

**}**

i**++;**

**}**

String ans**=**"digit="**+**d**+**"letter="**+**c**;**

out**.**println**(**ans**);**

**}**

out**.**close**();**

in**.**close**();**

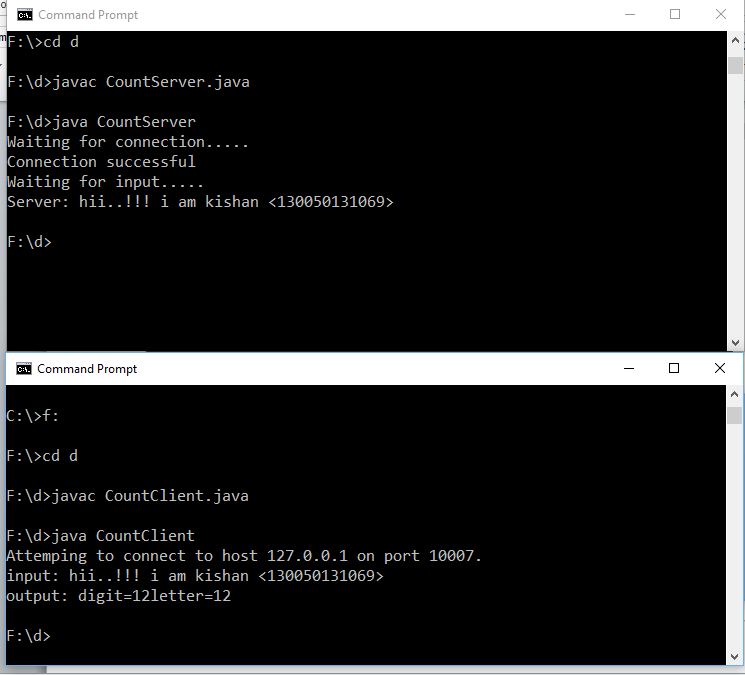
clientSocket**.**close**();**

serverSocket**.**close**();**

**}**

**}**

**Input Output:**

****

**(2)Connection-less Communication**

**UDP Client:**

**import** java**.**io**.\*;**

**import** java**.**net**.\*;**

class UDPClient **{**

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

System**.**out**.**println**(**"Eno=130050131069\n"**);**

**try{**

String serverHostname **=** **new** String **(**"127.0.0.1"**);**

**if** **(**args**.**length **>** 0**)**

serverHostname **=** args**[**0**];**

BufferedReader br **=** **new** BufferedReader**(new** InputStreamReader**(**System**.**in**));**

DatagramSocket ds **=** **new** DatagramSocket**();**

InetAddress IPAddress **=** InetAddress**.**getByName**(**serverHostname**);**

System**.**out**.**println **(**"Attemping to connect to " **+** IPAddress **+**") via UDP port 10009"**);**

byte**[]** sendData **=** **new** byte**[**256**];**

byte**[]** receiveData **=** **new** byte**[**256**];**

System**.**out**.**print**(**"Enter Message: "**);**

String sentence **=** br**.**readLine**();**

sendData **=** sentence**.**getBytes**();**

System**.**out**.**println **(**"Sending data to " **+** sendData**.**length **+** " bytes to server."**);**

DatagramPacket sendPacket **=** **new** DatagramPacket**(**sendData**,** sendData**.**length**,** IPAddress**,**10009**);**

ds**.**send**(**sendPacket**);**

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

System**.**out**.**println **(**"Waiting for return packet"**);**

ds**.**setSoTimeout**(**10000**);**

**try** **{**

ds**.**receive**(**receivePacket**);**

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

InetAddress returnIPAddress **=** receivePacket**.**getAddress**();**

int port **=** receivePacket**.**getPort**();**

System**.**out**.**println **(**"From server at: " **+** returnIPAddress **+**":" **+** port**);**

System**.**out**.**println**(**"Message: " **+** modifiedSentence**);**

**}**

**catch** **(**SocketTimeoutException ste**){**

System**.**out**.**println **(**"Timeout Occurred: Packet assumed lost"**);**

**}**

clientSocket**.**close**();**

**}**

**catch** **(**UnknownHostException ex**){**

System**.**err**.**println**(**ex**);**

**}**

**catch** **(**IOException ex**){**

System**.**err**.**println**(**ex**);**

**}**

**}**

**}**

**UDP Server:**

**import** java**.**io**.\*;**

**import** java**.**net**.\*;**

class UDPServer **{**

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

System**.**out**.**println**(**"Eno=130050131069\n"**);**

**try{**

DatagramSocket serverSocket **=** **new** DatagramSocket**(**10009**);**

byte**[]** receiveData **=** **new** byte**[**1024**];**

byte**[]** sendData **=** **new** byte**[**1024**];**

**while(true){**

receiveData **=** **new** byte**[**1024**];**

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

System**.**out**.**println **(**"Waiting for datagram packet"**);**

serverSocket**.**receive**(**receivePacket**);**

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

InetAddress IPAddress **=** receivePacket**.**getAddress**();**

int port **=** receivePacket**.**getPort**();**

System**.**out**.**println **(**"From: " **+** IPAddress **+** ":" **+** port**);**

System**.**out**.**println **(**"Message: " **+** sentence**);**

int i**=**0**,**d**=**0**,**c**=**0**;**

**while(**i**<=**sentence**.**length**()-**1**){**

**if** **(**Character**.**isDigit**(**sentence**.**charAt**(**i**))){**

d**++;**

**}**

**else** **if** **(**Character**.**isLetter**(**sentence**.**charAt**(**i**)))** **{**

c**++;**

**}**

i**++;**

**}**

String capitalizedSentence **=**"digit="**+**d**+**"letter="**+**c**;**

sendData **=** capitalizedSentence**.**getBytes**();**

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

serverSocket**.**send**(**sendPacket**);**

**}**

**}**

**catch** **(**SocketException ex**)** **{**

System**.**out**.**println**(**"UDP Port 10009 is occupied."**);**

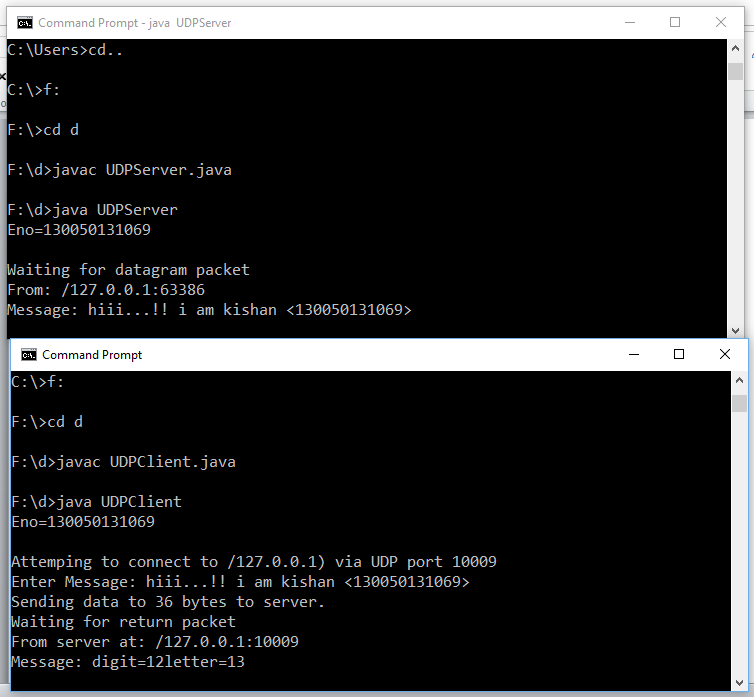
System**.**exit**(**1**);**

**}**

**}**

**}**

**Input Output:**

****