**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:**

**ClientTCP.java**

//TCP Client

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

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

public class ClientTCP **{**

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 10008."**);**

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

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

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

**try** **{**

echoSocket **=** **new** Socket**(**serverHostname**,** 10008**);**

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**);**

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

**}**

BufferedReader stdIn **=** **new** BufferedReader**(**

**new** InputStreamReader**(**System**.**in**));**

String userInput**;**

System**.**out**.**println **(**"Type String (\"Bye.\" to quit)"**);**

**while** **((**userInput **=** stdIn**.**readLine**())** **!=** **null)**

**{**

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

// end loop

**if** **(**userInput**.**equals**(**"Bye."**))**

**break;**

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

**}**

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

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

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

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

**}**

**}**

**ServerTCP.java**

//TCP Server

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

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

**import** java**.**util**.\*;**

public class ServerTCP **extends** Thread

**{**

protected Socket clientSocket**;**

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

**{**

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

**try** **{**

serverSocket **=** **new** ServerSocket**(**10008**);**

System**.**out**.**println**(**"Enroll: 130050131067"**);**

System**.**out**.**println **(**"Connection Socket Created"**);**

**try** **{**

**while** **(true)**

**{**

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

**new** ServerTCP **(**serverSocket**.**accept**());**

**}**

**}**

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

**{**

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

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

**}**

**}**

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

**{**

System**.**err**.**println**(**"Could not listen on port: 10008."**);**

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

**}**

**finally**

**{**

**try** **{**

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

**}**

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

**{**

System**.**err**.**println**(**"Could not close port: 10008."**);**

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

**}**

**}**

**}**

private ServerTCP**(**Socket clientSoc**)**

**{**

clientSocket **=** clientSoc**;**

start**();**

**}**

public void run**()**

**{**

System**.**out**.**println **(**"New Communication Thread Started"**);**

**try** **{**

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

**true);**

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

**new** InputStreamReader**(** clientSocket**.**getInputStream**()));**

String inputLine**;**

String**[]** inputTemp **=** **new** String**[**3**];**

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

**{**

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

int ans**[]** **=** **new** ProcessString**(**inputLine**).**process**();**

out**.**println**(**"No of characters is: "**+**ans**[**1**]+**" No of digits is: "**+**ans**[**0**]);**

**if** **(**inputLine**.**equals**(**"Bye."**))**

**break;**

**}**

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

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

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

**}**

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

**{**

System**.**err**.**println**(**"Problem with Communication Server"**);**

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

**}**

**}**

**}**

**ClientUDP.java**

//UDP Client

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

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

public class ClientUDP **{**

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

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**();**

BufferedReader stdIn **=** **new** BufferedReader**(**

**new** InputStreamReader**(**System**.**in**));**

String userInput**;**

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

**while** **((**userInput **=** stdIn**.**readLine**())** **!=** **null)** **{**

**if** **(**userInput**.**equals**(**"Bye."**))**

**break;**

byte**[]** b **=** userInput**.**getBytes**();**

InetAddress host **=** InetAddress**.**getByName**(**"localhost"**);**

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

host**.**getAddress**()** **+** " on port 10007."**);**

DatagramPacket dataPacket **=** **new** DatagramPacket**(**b**,** b**.**length**,** host**,** 10007**);**

datagramSocket**.**send**(**dataPacket**);**

System**.**out**.**println**(**"Client: "**+**userInput**);**

byte**[]** buffer **=** **new** byte**[**50**];**

DatagramPacket reply **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**reply**);**

String word **=** **new** String**(**reply**.**getData**());**

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

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

**}**

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

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

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

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

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

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

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

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

**}**

datagramSocket**.**close**();**

**}**

**}**

**ServerUDP.java**

//UDP Server

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

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

public class ServerUDP **{**

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

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**(**10007**);**

System**.**out**.**println**(**"Enroll: 130050131067"**);**

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

**while(true)** **{**

byte **[]** buffer **=** **new** byte**[**50**];**

DatagramPacket request **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**request**);**

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

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

String arrayMsg **=** **new** String**(**request**.**getData**());**

System**.**out**.**println**(**"From: " **+** request**.**getAddress**()** **+** " Port: " **+** request**.**getPort**());**

System**.**out**.**println**(**"Client: "**+**arrayMsg**);**

int ans**[]** **=** **new** ProcessString**(**arrayMsg**).**process**();**

String answer **=** "No of characters is: "**+**ans**[**1**]+**" No of digits is: "**+**ans**[**0**];**

byte**[]** sendMsg **=** answer**.**getBytes**();**

DatagramPacket reply **=** **new** DatagramPacket**(**sendMsg**,** sendMsg**.**length**,** request**.**getAddress**(),** request**.**getPort**());**

datagramSocket**.**send**(**reply**);**

String word **=** **new** String**(**reply**.**getData**());**

//System.out.println("Server: "+word);

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

**}**

**}catch(**Exception ex**)** **{**

**}**

datagramSocket**.**close**();**

**}**

**}**

**ProcessString.java**

//String Processing.

public class ProcessString **{**

String string**;**

ProcessString**(**String string**)** **{**

**this.**string **=** string**;**

**}**

public int**[]** process**()** **{**

int len **=** string**.**length**();**

int digits **=** 0**;**

int characters **=** 0**;**

int temp**[]** **=** **new** int**[**2**];**

**for(**int i **=** 0**;**i **<** string**.**length**();** i**++){**

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

digits**++;**

**}else** **if(**Character**.**isLetter**(**string**.**charAt**(**i**)))** **{**

characters**++;**

**}**

**}**

temp**[**0**]** **=** digits**;**

temp**[**1**]** **=** characters**;**

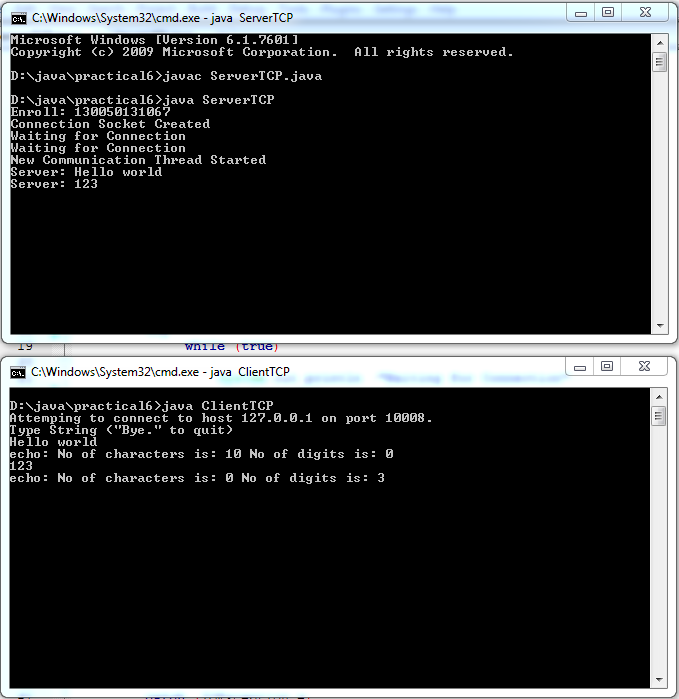
**return** temp**;**

**}**

**}**

**Input Output:**

For TCP



For UDP

