**Practical 1.2**

**Aim:**

Write a java program using java.net library concurrent echo client-server. (TCP Protocol)

**Description:**

****

Fig.1. Concurrent Client Server Scenario

**Program code:**

**Echo Client:**

//TCP Client

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

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

public class EchoClient2 **{**

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 Message (\"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**();**

**}**

**}**

**Echo Server:**

//TCP Server

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

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

public class EchoServer2 **extends** Thread

**{**

protected Socket clientSocket**;**

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

**{**

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

**try** **{**

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

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

**try** **{**

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

**{**

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

**new** EchoServer2 **(**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 EchoServer2 **(**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**;**

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

**{**

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

out**.**println**(**inputLine**);**

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

**break;**

**}**

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

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

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

**}**

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

**{**

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

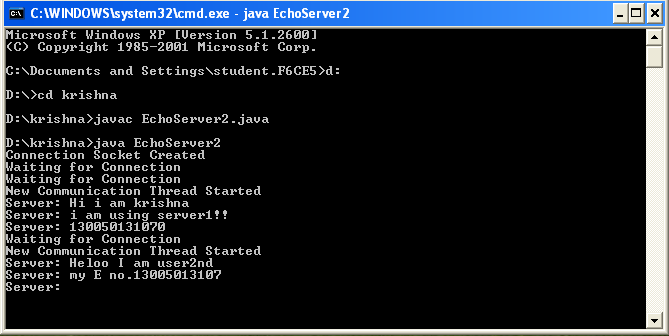
System**.**exit**(**1**);**

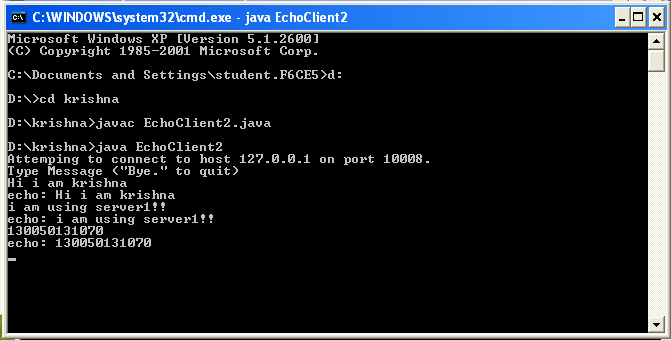
**}**

**}**

**}**

**Input Output:**

****

****

