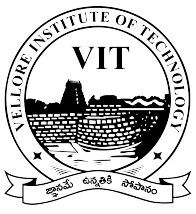
CSE3003 : Computer Networks

CN Lab 5 (Socket Programming)

Name	Gudi Varaprasad
Reg. No.	19BCE7048
Submitted to	Dr.R. NANDHA KUMAR sir
Lab Slot	L39 + L40
Department	School of Computer Science and Engineering
Email	varaprasad.19bce7048@vitap.ac.in
Date	24th February 2021



VIT®
AP

Question :

Implement a Simple Multi Client and server communication by using simple Arithmetic operations.

MultiServer.java

```
1  import java.util.*;
2  import java.io.*;
3  import java.net.*;
4
5  public class MultiServer {
6      public static void main(String[] args) throws Exception {
7          try {
8              // initialize server socket as 59768
9              ServerSocket server = new ServerSocket(59768);
10             // identify number of clients connected to the
                  server
11             int counter = 0;
12
13             System.out.println("Server Started .....");
14
15             while (true) {
16                 // increment when client is request
17                 counter++;
18                 // server accept the client connection by
                      binding
19                 Socket serverClient = server.accept();
```

```
20
21                  System.out.println(" >>> " + "Client No. : "
                        + counter + " started " + " <<< ");
22                  System.out.println();
23                  // Creating a thread to compute arthimetic
                        operations
24                  ServerClientThread sct = new
                        ServerClientThread(serverClient, counter);
25                  sct.start(); // starting the created thread
26              }
27          } catch (Exception e) {
28              // print all exceptions if any
29              System.out.println(e);
30          }
31      }
32  }
33
34  // IP address is : 192.168.100.9
35  // Host name : GVP
36  // Port used is : 59768
```

MultiClient.java

```
1
2   import java.util.*;
3   import java.io.*;
4   import java.net.*;
```

```
5
6  public class MultiClient {
7      public static void main(String[] args) throws Exception {
8          try {
9              // create socket with IP address and port number
                   of server
10             Socket socket = new Socket("192.168.100.9",
                   59768);
11             // getting options list add, sub, mul
12             DataInputStream option = new
                   DataInputStream(socket.getInputStream());
13             // sending information to server side
14             DataInputStream inStream = new
                   DataInputStream(socket.getInputStream());
15
16             // to send option to server to perform which
                   arthimetic operation
17             DataOutputStream outOption = new
                   DataOutputStream(socket.getOutputStream());
18             // for sending first value
19             DataOutputStream outStream1 = new
                   DataOutputStream(socket.getOutputStream());
20             // for sending second value
21             DataOutputStream outStream2 = new
                   DataOutputStream(socket.getOutputStream());
22
```

```
23          // for reading options in the list like add, sub,
               mul
24          BufferedReader br = new BufferedReader(new
               InputStreamReader(System.in));
25          // creating user input object for giving user
               inputs
26          BufferedReader num1 = new BufferedReader(new
               InputStreamReader(System.in));
27          BufferedReader num2 = new BufferedReader(new
               InputStreamReader(System.in));
28
29          // initialize reference variable in string format
30          String clientMessage = "", serverMessage = "",
               serverOutput = "", input1 = "", input2 = "";
31
32          // condition to check if the client message is
               END or not, to exit from loop
33          while (!clientMessage.equals("END")) {
34              System.out.println("Enter the option 1.Add \t
                   2.Subtract \t 3.Multiply \t 4.Divide \t
                   5.Square \t 6.Exit");
35              // read the options
36              clientMessage = br.readLine();
37              // send this option information to server
38              outOption.writeUTF(clientMessage);
39              // convert this information to UTF format
```

```
40              serverOutput = inStream.readUTF();
41              System.out.println(serverOutput);
42
43              System.out.print("Enter 1st number = ");
44              // Take user input number 1
45              input1 = num1.readLine();
46              System.out.println();
47              System.out.print("Enter 2nd number = ");
48              // Take user input number 2
49              input2 = num2.readLine();
50              System.out.println();
51              // pass these two inputs to the server side
52              outStream1.writeUTF(input1);
53              outStream2.writeUTF(input2);
54              // after performing operation, reading from
                   server side to client side
55              serverOutput = inStream.readUTF();
56              System.out.println(serverOutput);
57              // make buffer memory empty
58              outOption.flush();
59          }
60          // close all the connections
61          outOption.close();
62          outStream1.close();
63          outStream2.close();
64          socket.close();
```

```
65          } catch (Exception e) {
66              System.out.println(e);
67          }
68
69      }
70  }
71
72  // IP address is : 192.168.100.9
73  // Host name : GVP
74  // Port used is : 59768
```

ServerClientThread.java

```
1
2   import java.util.*;
3   import java.io.*;
4   import java.net.*;
5
6   // Concept of Multi threading
7   class ServerClientThread extends Thread {
8       Socket serverClient; // for creating a socket named
            serverClient
9       int clientNo; // client number
10
11      // constructor class to identify socket and which client
            request
12      ServerClientThread(Socket inSocket, int counter) {
```

```
13          // identify socket
14          serverClient = inSocket;
15          // which client request
16          clientNo = counter;
17      }
18
19      // running the thread created
20      public void run() {
21          try {
22              // get input option list from client using
                   serverClient socket
23              DataInputStream input1 = new
                   DataInputStream(serverClient.getInputStream());
24              // getting information to server side numbers
25              DataInputStream inStream1 = new
                   DataInputStream(serverClient.getInputStream());
26              DataInputStream inStream2 = new
                   DataInputStream(serverClient.getInputStream());
27              DataOutputStream outStream = new
                   DataOutputStream(serverClient.getOutputStream());
28
29              // initialize reference variable in string format
30              String clientInput1 = "", serverMessage = "",
                   clientInput2 = "", input = "", output = "";
31
32              // condition to check if the client message is
```

```
                END or not, to exit from loop
33              while (!clientInput1.equals("END")) {
34                  // verify the actual result of option input
35                  input = input1.readUTF();
36
37                  serverMessage = "From Server to Client - " +
                        clientNo;
38                  outStream.writeUTF(serverMessage);
39
40                  // Input number 1 from client
41                  clientInput1 = inStream1.readUTF();
42                  // Input number 2 from client
43                  clientInput2 = inStream2.readUTF();
44
45                  int optionNo, sum, difference, product,
                        division, square;
46                  // which option to select
47                  optionNo = Integer.parseInt(input);
48                  // parses a string (UTF here) and converts it
                        to an integer
49
50                  // check option is 1 or not means to perform
                        sumition
51                  if (optionNo == 1) {
52                      System.out.println("From Client - " +
                            clientNo + " : Number1 is : " +
```

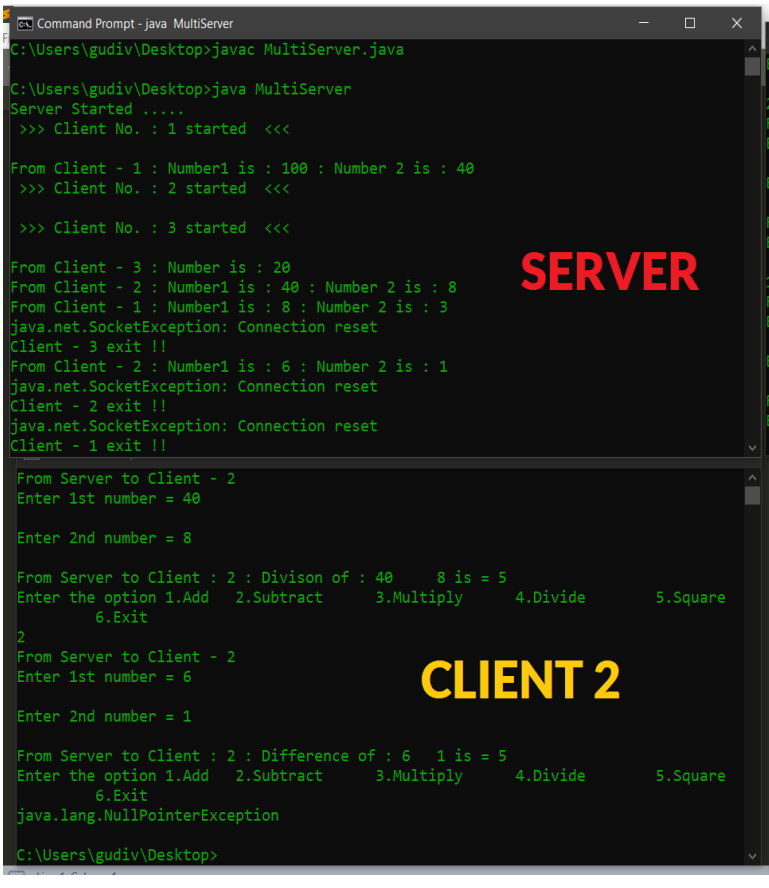
```
                clientInput1 + " : Number 2 is : " +
                clientInput2);
53              sum = Integer.parseInt(clientInput1) +
                Integer.parseInt(clientInput2);
54              output = "From Server to Client : " +
                clientNo + " : Sum of : " +
                clientInput1 + "\t" + clientInput2 + "
                is = " + sum;
55              outStream.writeUTF(output);
56          }
57          // else check option is 2 or not means to
              perform differencetraction
58          else if (optionNo == 2) {
59              System.out.println("From Client - " +
                clientNo + " : Number1 is : " +
                clientInput1 + " : Number 2 is : " +
                clientInput2);
60              difference =
                Integer.parseInt(clientInput1) -
                Integer.parseInt(clientInput2);
61              output = "From Server to Client : " +
                clientNo + " : Difference of : " +
                clientInput1 + "\t" + clientInput2 + "
                is = " + difference;
62              outStream.writeUTF(output);
63          }
```

```
64                  // else check option is 3 or not means to
                       perform producttiplication
65                  else if (optionNo == 3) {
66                      System.out.println("From Client - " +
                           clientNo + " : Number1 is : " +
                           clientInput1 + " : Number 2 is : " +
                           clientInput2);
67                      product = Integer.parseInt(clientInput1) *
                           Integer.parseInt(clientInput2);
68                      output = "From Server to Client : " +
                           clientNo + " : Product of : " +
                           clientInput1 + "\t" + clientInput2 + "
                           is = " + product;
69                      outStream.writeUTF(output);
70                  }
71                  // else check option is 4 or not means to
                       perform divisionision
72                  else if (optionNo == 4) {
73                      System.out.println("From Client - " +
                           clientNo + " : Number1 is : " +
                           clientInput1 + " : Number 2 is : " +
                           clientInput2);
74                      division = Integer.parseInt(clientInput1)
                           / Integer.parseInt(clientInput2);
75                      output = "From Server to Client : " +
                           clientNo + " : Divison of : " +
```

```
                    clientInput1 + "\t" + clientInput2 + "
                    is = " + division;
76                  outStream.writeUTF(output);
77              }
78              // else check option is 5 or not means to
                   perform squareuare
79              else if (optionNo == 5) {
80                  System.out.println("From Client - " +
                        clientNo + " : Number is : " +
                        clientInput1);
81                  square = Integer.parseInt(clientInput1) *
                        Integer.parseInt(clientInput1);
82                  output = "From Server to Client : " +
                        clientNo + " : Square of : " +
                        clientInput1 + " is = " + square;
83                  outStream.writeUTF(output);
84              }
85              // neither of them, default option to exit
86              else {
87                  System.exit(0);
88              }
89              // make buffer memory empty
90              outStream.flush();
91          }
92          // close all the connections
93          inStream1.close();
```

```
94              outStream.close();
95              serverClient.close();
96          } catch (Exception e) {
97              // print if any exceptions
98              System.out.println(e);
99          } finally {
100             // default method to print which client got
                   diconnected
101             System.out.println("Client - " + clientNo + "
                   exit !!");
102         }
103     }
104 }
105
106 // IP address is : 192.168.100.9
107 // Host name : GVP
108 // Port used is : 59768
```

Output



If exceptions ?