

I. Develop a client server application where a client types a message and presses the “Enter” key to send it to the server. The server responds to the message. Client should be prevented from sending the next message until it receives the response for the previous message. Your implementation should support not less than 3 clients active at the same time. This may be implemented either by using connectionless or connection oriented sockets.

Details of the clients connected such as IP address, port number have to be displayed at the server side.

Server:

```
package endsem;

import java.io.*;
import java.text.*;
import java.util.*;
import java.net.*;

public class Server
{
    public static void main(String[] args) throws IOException
    {
        ServerSocket ss = new ServerSocket(5000);

        while (true)
        {
            Socket s = null;

            try
            {
                s = ss.accept();

                System.out.println("Client connected: " + s);

                DataInputStream dis = new DataInputStream(s.getInputStream());
                DataOutputStream dos = new DataOutputStream(s.getOutputStream());

                Thread t = new ClientHandler(s, dis, dos);

                t.start();
            }
            catch (Exception e){
                s.close();
                e.printStackTrace();
            }
        }
    }
}

class ClientHandler extends Thread
{
    final DataInputStream dis;
    final DataOutputStream dos;
    final Socket s;

    public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos)
    {
        this.s = s;
        this.dis = dis;
        this.dos = dos;
    }

    @Override
    public void run()
```

```

{
    String received;
    String toreturn;
    while (true)
    {
        try {

            received = dis.readUTF();

            dos.writeUTF("Received message");
            System.out.println(received);

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
}

```

The screenshot shows an IDE with three windows: **Server.java**, **Client.java**, and three **Console** windows.

Server.java code:

```

1 package endsem;
2
3 import java.io.*;
4 import java.text.*;
5 import java.util.*;
6 import java.net.*;
7
8 public class Server
9 {
10     public static void main(String[] args) throws IOException
11     {
12         ServerSocket ss = new ServerSocket(5000);
13
14         while (true)
15         {
16             Socket s = null;
17
18             try
19             {
20                 s = ss.accept();
21
22                 System.out.println("Client connected: " + s);
23
24                 DataInputStream dis = new DataInputStream(s.getInputStream());
25                 DataOutputStream dos = new DataOutputStream(s.getOutputStream());
26
27                 Thread t = new ClientHandler(s, dis, dos);
28
29                 t.start();
30
31             }
32             catch (Exception e){
33                 s.close();
34                 e.printStackTrace();
35             }
36         }
37     }
38 }
39
40 class ClientHandler extends Thread
41 {
42     final DataInputStream dis;
43     final DataOutputStream dos;

```

Client.java code (partially visible):

```

1 package endsem;
2
3 import java.io.*;
4 import java.text.*;
5 import java.util.*;
6 import java.net.*;
7
8 public class Client
9 {
10     public static void main(String[] args) throws IOException
11     {
12         Socket s = new Socket("localhost", 5000);
13
14         DataInputStream dis = new DataInputStream(s.getInputStream());
15         DataOutputStream dos = new DataOutputStream(s.getOutputStream());
16
17         while (true)
18         {
19             String message = "client1";
20             dos.writeUTF(message);
21             String received = dis.readUTF();
22             System.out.println("Received message: " + received);
23         }
24     }
25 }

```

Console 1 (Server):

```

Server [Java Application] C:\Program Files\Java\jre\bin\java.exe
Client connected: Socket[addr=/127.0.0.1,port=5000]
client1
client2
client3
client2
client3
client1

```

Console 2 (Client 1):

```

Client [Java Application] C:\Program Files\Java\jre\bin\java.exe
client1
Received message
client1
Received message

```

Console 3 (Client 2):

```

Client [Java Application] C:\Program Files\Java\jre\bin\java.exe
client2
Received message
client2
Received message

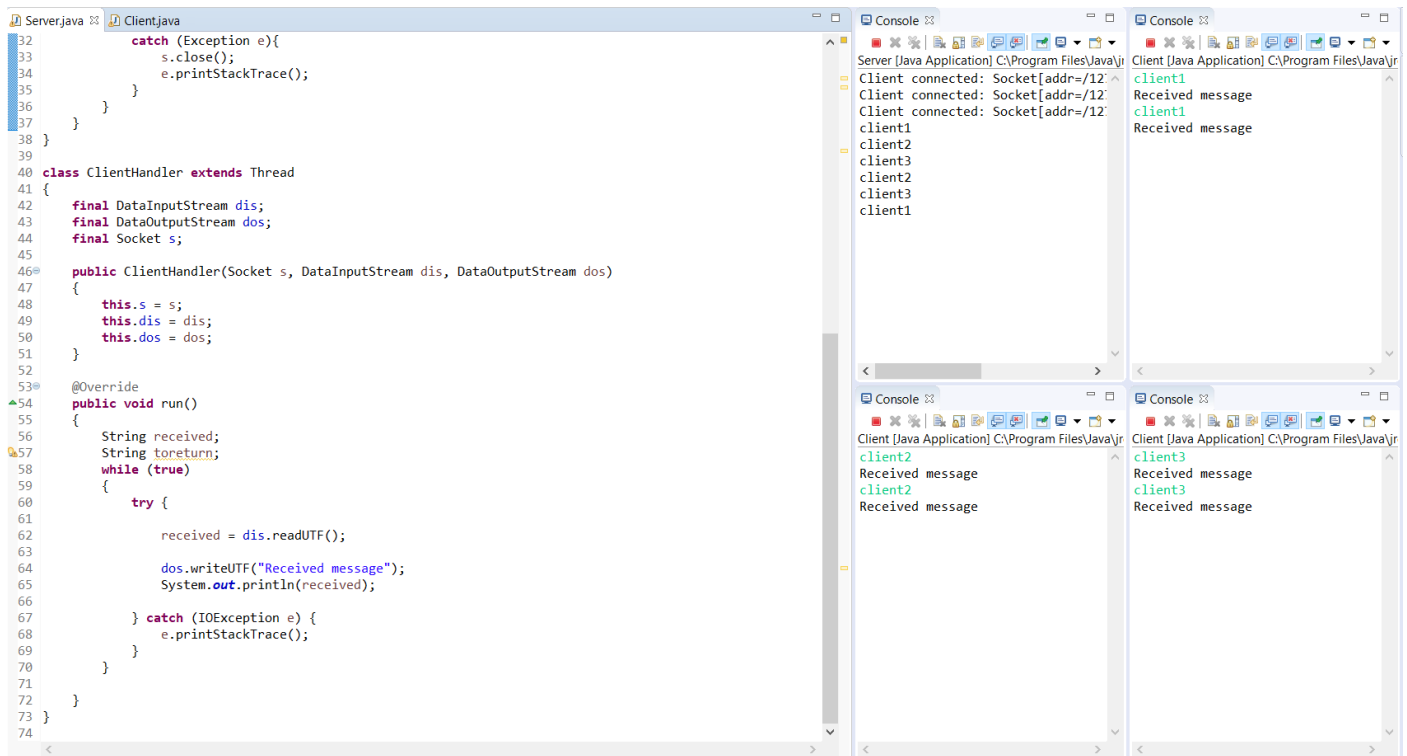
```

Console 4 (Client 3):

```

Client [Java Application] C:\Program Files\Java\jre\bin\java.exe
client3
Received message
client3
Received message

```



Client:

```

package endsem;

import java.io.*;
import java.net.*;
import java.util.Scanner;

public class Client
{
    public static void main(String[] args) throws IOException
    {
        try
        {
            Scanner scn = new Scanner(System.in);

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

            Socket s = new Socket(ip, 5000);

            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());

            while (true)
            {
                String tosend = scn.nextLine();
                dos.writeUTF(tosend);

                System.out.println(dis.readUTF());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}
}

```

```

Server.java
1 package endsem;
2
3 import java.io.*;
4 import java.net.*;
5 import java.util.Scanner;
6
7 public class Client
8 {
9     public static void main(String[] args) throws IOException
10    {
11        try
12        {
13            Scanner scn = new Scanner(System.in);
14
15            InetAddress ip = InetAddress.getByName("localhost");
16
17            Socket s = new Socket(ip, 5000);
18
19            DataInputStream dis = new DataInputStream(s.getInputStream());
20            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
21
22            while (true)
23            {
24                String tosend = scn.nextLine();
25                dos.writeUTF(tosend);
26
27                System.out.println(dis.readUTF());
28            }
29        } catch (Exception e) {
30            e.printStackTrace();
31        }
32    }
33 }
34
35
Client.java
1 package endsem;
2
3 import java.io.*;
4 import java.net.*;
5 import java.util.Scanner;
6
7 public class Client
8 {
9     public static void main(String[] args) throws IOException
10    {
11        try
12        {
13            Scanner scn = new Scanner(System.in);
14
15            InetAddress ip = InetAddress.getByName("localhost");
16
17            Socket s = new Socket(ip, 5000);
18
19            DataInputStream dis = new DataInputStream(s.getInputStream());
20            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
21
22            while (true)
23            {
24                String tosend = scn.nextLine();
25                dos.writeUTF(tosend);
26
27                System.out.println(dis.readUTF());
28            }
29        } catch (Exception e) {
30            e.printStackTrace();
31        }
32    }
33 }
34
35
Console
Server [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (19-Jun-2020, 4:32:15 PM)
Client connected: Socket[addr=/127.0.0.1,port=59464,localport=5000]
Client connected: Socket[addr=/127.0.0.1,port=59465,localport=5000]
Client connected: Socket[addr=/127.0.0.1,port=59467,localport=5000]
client1
client2
client3
client2
client3
client1

```

Output:

```

Console
Server [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (19-Jun-2020, 4:32:15 PM)
Client connected: Socket[addr=/127.0.0.1,port=59464,localport=5000]
Client connected: Socket[addr=/127.0.0.1,port=59465,localport=5000]
Client connected: Socket[addr=/127.0.0.1,port=59467,localport=5000]
client1
client2
client3
client2
client3
client1

Console
Client [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (19-Jun-2020, 4:32:33 PM)
client1
Received message
client1
Received message

Console
Client [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (19-Jun-2020, 4:32:33 PM)
client2
Received message
client2
Received message

Console
Client [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (19-Jun-2020, 4:32:34 PM)
client3
Received message
client3
Received message

```

II. Fill in the below table for an VLSM network of class C address:192.168.10.0/24 to allocate. Cities A,B have a WAN connection to City C. The requirements for each city are

City A requires 60 hosts.

City B requires 28 hosts.

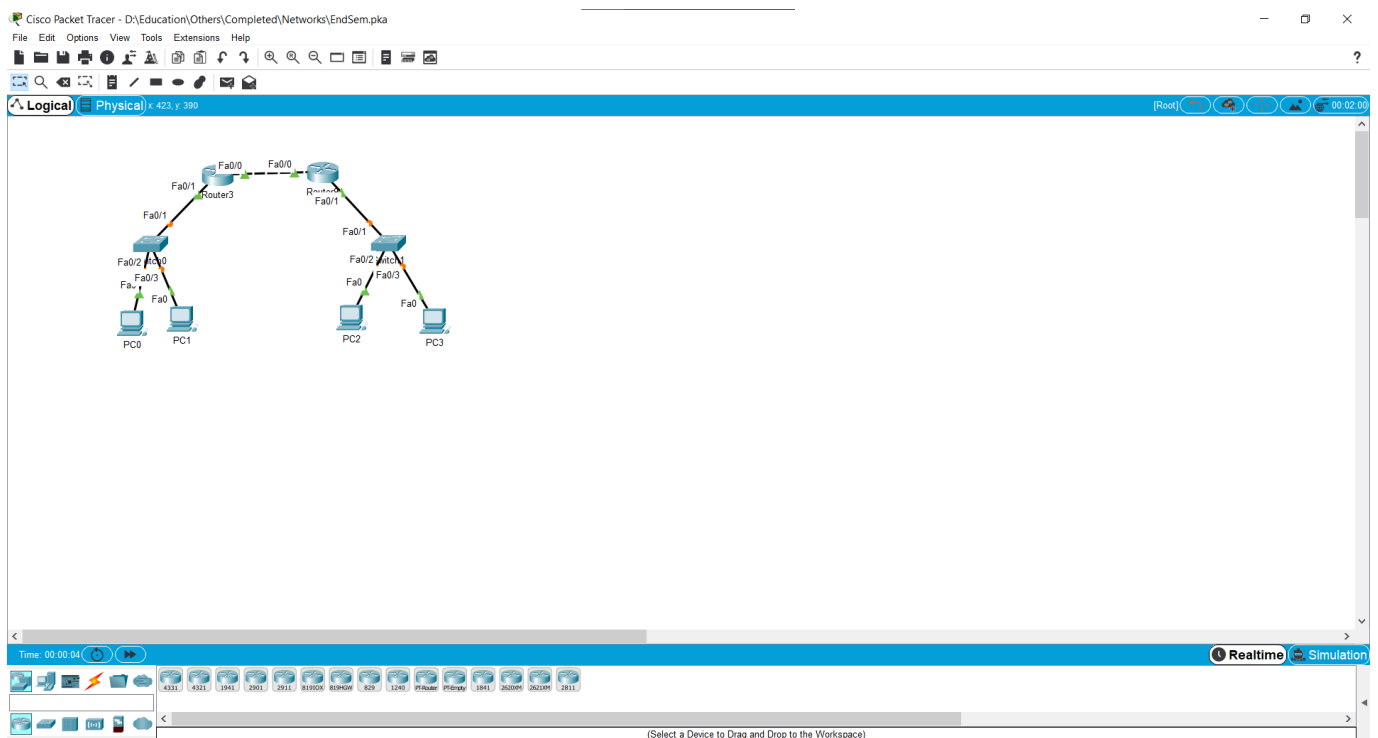
City C requires 12 hosts.

The addresses for the WAN links can be taken from the remaining usable addresses available

Subnet	Subnet Address	Subnet Mask(/x)	First Usable Host	Last Usable Host	Broadcast
City A	255.255.255.192	/26	192.168.10.1	192.168.10.62	192.168.10.63
City B	255.255.255.224	/27	192.168.10.65	192.168.10.94	192.168.10.95
City C	255.255.255.240	/28	192.168.10.97	192.168.10.110	192.168.10.111
WAN1 (A to C)	255.255.255.252	/30	192.168.10.113	192.168.10.114	192.168.10.115
WAN2 (B to C)	255.255.255.252	/30	192.168.10.117	192.168.10.118	192.168.10.119

III. Design using packet tracer to connect any above two networks. Assign the first and last usable address to each city in the initial network.(Open the Initial network named -EndSem.pka file). Show the successful communication for

Source	Destination	Type	
PC0	PC1 : 0.0.0.0	ICMP	
PC2	PC3 : 0.0.0.0	ICMP	
PC0	Router3 : 0.0.0.0	ICMP	
PC1	Router9 : 0.0.0.0	ICMP	
PC2	Router9 : 0.0.0.0	ICMP	
PC3	Router3 : 0.0.0.0	ICMP	
PC0	PC2 : 0.0.0.0	ICMP	
PC3	PC1 : 0.0.0.0	ICMP	



Cisco Packet Tracer - D:\Education\Others\Completed\Networks\EndSem.pka

File Edit Options View Tools Extensions Help

Activity Results Time Left: 00:50:06

Congratulations Guest! You completed the activity.

[Overall Feedback](#) [Assessment Items](#) [Connectivity Tests](#)

Congratulations on completing this activity!

[Close](#)

Activity Results Time Left: 00:49:11

Congratulations Guest! You completed the activity.

[Overall Feedback](#) [Assessment Items](#) [Connectivity Tests](#)

[Expand/Collapse All](#) [Show Incorrect Items](#)

Assessment Item	Status	Points	Component(s)	Feedback
✓ Network	Correct	0	Other	

Score : 20/20

Item Count : 9/9

Component	Items/Total	Score
Connectivity	8/8	20/20
Connectivity Tests	8/8	20/20

[Close](#)

Cisco Packet Tracer - D:\Education\Others\Completed\Networks\EndSem.pka

File Edit Options View Tools Extensions Help

Activity Results Time Left: 00:48:46

Congratulations Guest! You completed the activity.

[Overall Feedback](#) [Assessment Items](#) [Connectivity Tests](#)

Below are the results of your connectivity tests:

	Status	Test Condition	Points	Source	Destination	Type
1	Correct	Successful	1	PC0	PC1 : 192.168.10.62	ICMP
2	Correct	Successful	1	PC2	PC3 : 192.168.10.110	ICMP
3	Correct	Successful	2	PC0	Router3 : 192.168.10.113	ICMP
4	Correct	Successful	3	PC1	Router9 : 192.168.10.114	ICMP
5	Correct	Successful	2	PC2	Router9 : 192.168.10.114	ICMP
6	Correct	Successful	3	PC3	Router3 : 192.168.10.113	ICMP
7	Correct	Successful	4	PC0	PC2 : 192.168.10.97	ICMP
8	Correct	Successful	4	PC3	PC1 : 192.168.10.62	ICMP
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

[Close](#)