DCN Assignment - 1

3152 Hardik Togadiya

1. **Write a java program to implement Client-Server Chat Application using TCP, in which when the client writes “bye”, then only the connection gets closed.**

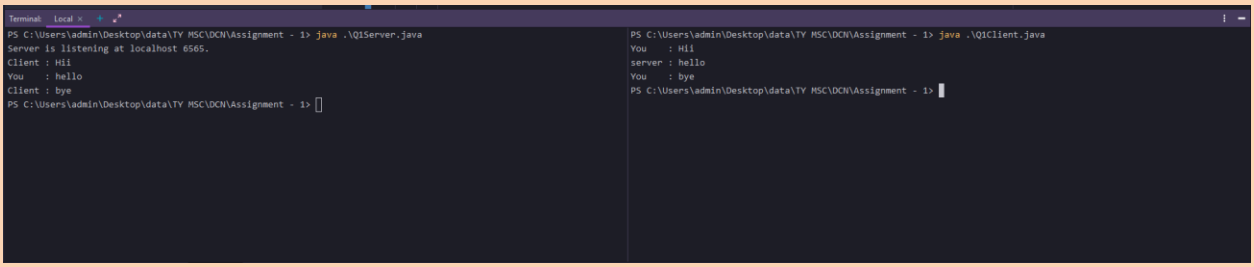
Q1Server.java

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

class Q1Server {
    public static void main(String[] args) throws Exception {
        try {
            ServerSocket ss = new ServerSocket(6565);
            System.out.println("Server is listening at localhost 6565.");

            Socket s = ss.accept();

            DataInputStream is = new DataInputStream(s.getInputStream());
            DataOutputStream os = new DataOutputStream(s.getOutputStream());
            String clientMsg = "", serverMsg = "";
            Scanner s1 = new Scanner(System.in);

            while (true) {
                clientMsg = is.readUTF();
                System.out.println("Client : " + clientMsg);

                if (clientMsg.equals("bye"))
                    break;

                System.out.print("You    : ");
                serverMsg = s1.nextLine();
                os.writeUTF(serverMsg);
                os.flush();
            }
            os.close();
            is.close();
            s.close();
            ss.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Q1Client.java

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

class Q1Client {
    public static void main(String[] args) throws Exception {
        try {
            Socket s = new Socket("localhost", 6565);

            DataInputStream is = new DataInputStream(s.getInputStream());
            DataOutputStream os = new DataOutputStream(s.getOutputStream());
            String clientMsg = "", serverMsg = "";
            Scanner s1 = new Scanner(System.in);

            while (true) {
                System.out.print("You    : ");
                clientMsg = s1.nextLine();
                os.writeUTF(clientMsg);
                os.flush();

                if (clientMsg.equals("bye")) {
                    break;
                }


                serverMsg = is.readUTF();
                System.out.println("server : " + serverMsg);


            }
            os.close();
            is.close();
            s.close();

        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

output:



2. Write a java program to Send and Receive Messages between two parties using UDP.

Q2Server.java

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

class Q2Server{
    public static void main(String[] args) throws Exception
    {
        try
        {
            DatagramSocket ds1 = new DatagramSocket(6565);
            System.out.println("Server is listening at localhost 6565.");

            byte[] buf = new byte[50];
            DatagramPacket dp1 = new DatagramPacket(buf,50);
            ds1.receive(dp1);
            String msg = new String(buf);
            System.out.println("Client : "+msg);

            DatagramSocket ds2 = new DatagramSocket();
            String msg1 = "Hello, This Message is from server side.";
            InetAddress ip = InetAddress.getByName("localhost");
            DatagramPacket dp2 = new
DatagramPacket(msg1.getBytes(),msg1.length(),ip,6666);
            ds2.send(dp2);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
```

```
    }
}
```

Q2Client.java

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

class Q2Client{
    public static void main(String[] args) throws Exception
    {
        try
        {
            DatagramSocket ds1 = new DatagramSocket();

            String msg="Hello, This Message is from client side.";
            InetAddress ip = InetAddress.getByName("localhost");
            DatagramPacket dp1 = new
DatagramPacket(msg.getBytes(),msg.length(),ip,6565);
            ds1.send(dp1);

            DatagramSocket ds2 = new DatagramSocket(6666);
            byte[] buf = new byte[50];
            DatagramPacket dp2 = new DatagramPacket(buf,50);
            ds2.receive(dp2);

            String str = new String(buf);
            System.out.println("Server : "+str);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

Server :

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q2Server.java
Server is listening at localhost 6565.
Client : Hello, This Message is from client side.
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

Client :

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q2Client.java
Server : Hello, This Message is from server side.
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

3. Write a java program to create an Echo Server using TCP, in which whatever message the client writes to the server, the server replies back with the same message.

Q3Server.java

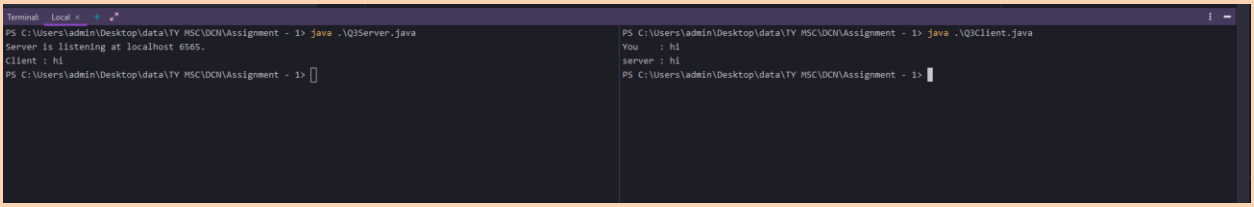
```java
import java.io.*;
import java.net.*;

class Q3Server{
    public static void main(String[] args) throws Exception
    {
        try
        {
            ServerSocket ss = new ServerSocket(6565);
            System.out.println("Server is listening at localhost 6565.");

            Socket s =ss.accept();

            String clientmsg="";
            DataInputStream is = new DataInputStream(s.getInputStream());
            clientmsg = is.readUTF();
            System.out.println("Client : "+clientmsg);


            String servermsg = clientmsg;
            DataOutputStream os = new DataOutputStream(s.getOutputStream());
            os.writeUTF(servermsg);
            os.flush();


            os.close();
            is.close();
            s.close();
            ss.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
```

```
    }
}
```

Q3Client.java

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

class Q3Client{
    public static void main(String[] args) throws Exception
    {
        try
        {
            Socket s = new Socket("localhost",6565);

            String clientmsg="";
            Scanner s1 = new Scanner(System.in);
            DataOutputStream os = new DataOutputStream(s.getOutputStream());
            System.out.print("You    : ");
            clientmsg = s1.nextLine();
            os.writeUTF(clientmsg);
            os.flush();

            String servermsg="";
            DataInputStream is = new DataInputStream(s.getInputStream());
            servermsg = is.readUTF();
            System.out.println("server : "+servermsg);


            os.close();
            is.close();
            s.close();

        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

OutPut :



4. Write a java program to demonstrate UDP implementation, in which client sends a number to the server and the server calculates the Cube of that number and sends back the result.

Q4Server.java

```
import java.net.*;
class Q4Server
{
    public static void main(String args[]) throws Exception
    {
        DatagramSocket ds=new DatagramSocket(9999);
        System.out.println("Server started");
        byte b1[]=new byte[1024];
        DatagramPacket dp=new DatagramPacket(b1,b1.length);
        ds.receive(dp);

        String str=new String(dp.getData());
        int num=Integer.parseInt(str.trim());
        int ans=num*num*num;

        byte b[]=(ans+"").getBytes();
        InetAddress ia=InetAddress.getLocalHost();
        DatagramPacket dp1=new DatagramPacket(b,b.length,ia,dp.getPort());
        ds.send(dp1);

        System.out.println("Result sent");
    }
}
```

Q4Client.java

```
import java.net.*;
import java.util.*;
class Q4Client
{
    public static void main(String args[]) throws Exception
```

```
    {
        DatagramSocket ds=new DatagramSocket();
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter Number");
        int i=sc.nextInt();

        byte b[]=(i+"").getBytes();

        InetAddress ia=InetAddress.getLocalHost();
        DatagramPacket dp=new DatagramPacket(b,b.length,ia,9999);
        ds.send(dp);

        byte b1[]=new byte[1024];
        DatagramPacket dp1=new DatagramPacket(b1,b1.length);
        ds.receive(dp1);

        String str=new String(dp1.getData());
        System.out.println("Result"+str.trim());
    }
}
```

Output :

Server

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q4Server.java
Server started
Result sent
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

Client

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q4Client.java
Enter Number
3
Result27
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

5. Write a java program to create a Multithreaded TCP Server which can handle more than one client at the same time. Create the Client-Server Chat Application.

Q5Server.java

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

class Q5Server{
    public static void main(String[] args) throws Exception
    {
        int counter=0;
        try
        {
            ServerSocket ss = new ServerSocket(6565);
            System.out.println("Server is listening at localhost 6565.");

            while(true)
            {
                Socket s= ss.accept();
                counter++;
                System.out.println("\nNew Client-"+ counter +" Connected.");

                clientThread cs = new clientThread(s,counter);
                cs.start();
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }

    }
}
class clientThread extends Thread{
    Socket cs = null;
    int clientno;

    clientThread(Socket s, int counter)
    {
        cs=s;
        clientno=counter;
    }
    public void run()
    {
```

```
        int counter = 0;
        try{
            DataInputStream is = new DataInputStream(cs.getInputStream());
            DataOutputStream os = new DataOutputStream(cs.getOutputStream());
            String clientmsg="",servermsg="";
            Scanner s1 = new Scanner(System.in);

            while(true)
            {

                clientmsg = is.readUTF();
                System.out.println("Client : "+clientmsg);

                if(clientmsg.equals("bye"))
                {
                    break;
                }

                System.out.print("You : ");
                servermsg = s1.nextLine();
                os.writeUTF(servermsg);
                os.flush();
            }
            cs.close();
            os.close();
            is.close();
            //s.close();
            //ss.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

Q5Client.java

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

class Q5Client{
    public static void main(String[] args) throws Exception
    {
        try
        {
            Socket s = new Socket("localhost",6565);

            DataInputStream is = new DataInputStream(s.getInputStream());
            DataOutputStream os = new DataOutputStream(s.getOutputStream());
```

```
            String clientmsg="",servermsg="";
            Scanner s1 = new Scanner(System.in);

            while(true)
            {
                System.out.print("You    : ");
                clientmsg = s1.nextLine();
                os.writeUTF(clientmsg);
                os.flush();

                if(clientmsg.equals("bye"))
                {
                    break;
                }


                servermsg = is.readUTF();
                System.out.println("server : "+servermsg);


            }
            os.close();
            is.close();
            s.close();

        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q5Client.java
You    : 100
server : 100
You    : hey
server : y
You    : bye
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q5Server.java
Server is listening at localhost 6565.

New Client-1 Connected.
Client : 100
You : 100
Client : hey
You : y
Client : bye
```

6. Write a java program to implement Byte Stuffing using TCP.

Q6Server.java

```
import java.net.*;
import java.util.*;
import java.io.*;
class Q6Server
{
    public static void main(String args[])throws Exception
    {
        ServerSocket ss= new ServerSocket(9999);
        System.out.println("Server started");
        Socket socket=ss.accept();
        System.out.println("Request accepted");
        InputStream dis=socket.getInputStream();
        byte b[]=new byte[1024];
        dis.read(b);

        String s= new String(b);
        System.out.println("Stuffed data from client:"+s.trim());

        System.out.println("Unstuffed data:");
        String str="";

        for(int i=1;i<(s.trim()).length()-1;i++)
        {
            char ch=s.charAt(i);
            if(ch=='E'||ch=='F')
            {
                ch=s.charAt(i+1);
                str+=ch;
                i++;
            }
            else
            {
```

```
                str+=ch;
            }
        }
        System.out.println(str);
    }
}
```

Q6Client.java

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
class Q6Client
{
    public static void main(String args[])throws Exception
    {
        Socket socket=new Socket("localhost",9999);
        InputStream dis= socket.getInputStream();
        OutputStream dos= socket.getOutputStream();

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter data:");
        String data=sc.next();

        String str="";
        String str1="F";
        for(int i=0;i<data.length();i++)
        {
            char ch=data.charAt(i);
            if(ch=='E'||ch=='F')
            {
                str+='E';
                str+=data.charAt(i);
            }
            else
            {
                str+=data.charAt(i);
            }
        }
        String Final=str1+" "+str+" "+str1;

        System.out.println("Data stuffed in client:"+Final);
        System.out.println("sending to server for unstuffing");
        dos.write(Final.getBytes());
    }
}
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q6Server.java
Server started
Request accepted
Stuffed data from client:F 100 F
Unstuffed data:
 100
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q6Client.java
Enter data:
100
Data stuffed in client:F 100 F
sending to server for unstuffing
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

7. Write a java program to implement Bit Stuffing using TCP.

Q7Server.java

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

public class Q7Server
{
    public static void main(String[] args) throws IOException
    {
        ServerSocket sr=new ServerSocket(9999);
        System.out.println("Server started");
        Socket skt=sr.accept();
        System.out.println("Request accepted");

        InputStream in=skt.getInputStream();
        byte b[]=new byte[1024];
        in.read(b);

        String str=new String(b);
        str=str.trim();
        System.out.println("Data from client "+str);
        int cnt=0;
        String s="";
        for(int i=8;i<str.length()-8;i++)
        {
            char c=str.charAt(i);
```

```
            if(c=='1')
            {
                cnt++;
                if(cnt==5)
                {
                    i++;
                    cnt=0;
                    s=s+c;
                }
                else
                {
                    s=s+c;
                }
            }
            else
            {
                s=s+c;
            }
        }
        System.out.println("Unstuffed data "+s);
    }
}
```

Q7Client.java

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

public class Q7Client
{
    public static void main(String[] args) throws IOException
    {
        Socket skt=new Socket("localhost",9999);

        InputStream in=skt.getInputStream();
        OutputStream out=skt.getOutputStream();

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter data");
        String data=sc.nextLine();
        data=data.trim();
        int cnt=0;
        String s="";
        for(int i=0;i<data.length();i++)
        {
            char c=data.charAt(i);
            if(c=='1')
            {
                cnt++;
                if(cnt<5)
```

```
                {
                    s=s+c;
                }
                else
                {
                    s=s+c+'0';
                    cnt=0;
                }
            }
            else
            {
                s=s+c;
                cnt=0;
            }
        }
        s="01111110"+" "+s+" "+"01111110";

        System.out.println("Data stuffed in client "+s);
        System.out.println("Sending data to unstuff");
        out.write(s.getBytes());
    }
}
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q7Server.java
Server started
Request accepted
Data from client 01111110 hello 01111110
Unstuffed data  hello
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q7Server.java
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q7Client.java
Enter data
hello
Data stuffed in client 01111110 hello 01111110
Sending data to unstuff
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

8. Write a java program to implement Character Count using UDP.

Q8Server.java

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

public class Q8Server
{
    public static void main(String[] args) throws IOException
    {
        DatagramSocket ds=new DatagramSocket(9999);
        System.out.println("Server started");
        byte b1[]=new byte[1024];
        DatagramPacket dp=new DatagramPacket(b1,b1.length);
        ds.receive(dp);

        String str=new String(dp.getData());
        str=str.trim();
        System.out.println("Received from client "+str);
        int length = str.length() + 1;
        String str1=length+str;

        System.out.println("Sending to client "+str1);
        byte b[]=(str1+"").getBytes();
        InetAddress ia=InetAddress.getLocalHost();
        DatagramPacket dp1=new DatagramPacket(b,b.length,ia,dp.getPort());
        ds.send(dp1);
        System.out.println("Data sent");
    }
}
```

Q8Client.java

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

public class Q8Client
{
    public static void main(String[] args) throws IOException
    {
        DatagramSocket ds=new DatagramSocket();
        String str;
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter String");
        str=sc.nextLine();

        byte b[]=(str+"").getBytes();

        InetAddress ia=InetAddress.getLocalHost();
        DatagramPacket dp=new DatagramPacket(b,b.length,ia,9999);
```

```
        ds.send(dp);

        byte b1[]=new byte[1024];
        DatagramPacket dp1=new DatagramPacket(b1,b1.length);
        ds.receive(dp1);

        String str1=new String(dp1.getData());
        str1=str1.trim();
        System.out.println("Received From Server " +str1);
        if(str1.length()<10)
        {
            System.out.println("Original data:"+str1.substring(1));
        }
        else
        {
            System.out.println("Original data:"+str1.substring(2));
        }
    }
}
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q8Client.java
Enter String hello
Received From Server 6hello
Original data:hello
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```

```
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1> java .\Q8Server.java
Server started
Received from client hello
Sending to client 6hello
Data sent
PS C:\Users\admin\Desktop\data\TY MSC\DCN\Assignment - 1>
```