

# Computer Networks

## Assignment 3

Name: Krunal Rank

Roll No: U18CO081

Client:

```
1  import java.io.*;
2  import java.net.*;
3
4  public class Client{
5
6      private DataInputStream input = null;
7      private DataOutputStream out = null;
8      private DataInputStream in = null;
9
10     public Client(String address, int port){
11
12         Socket socket = null;
13         try{
14             socket = new Socket(address, port);
15             System.out.println("Connected");
16             input = new DataInputStream(System.in);
17             out = new DataOutputStream(socket.getOutputStream());
18             in = new DataInputStream(new BufferedInputStream(socket.getInputStream()));
19         }catch(UnknownHostException u){
20             System.out.println("Unable to connect! Unknown host!");
21         }catch(IOException i){
22             System.out.println("Invalid Input by User!");
23         }
24
25         Thread readMessage = new Thread(new Runnable() {
26
27             @Override
28             public void run() {
29
30                 while (true) {
31                     try {
32
33                         // read the message sent to this client
34                         String msg = in.readUTF();
35                         System.out.println("Server: "+ msg);
36                     } catch (IOException e) {
37                         System.out.println(
38                             "Unknown Error Occured while reading Message! Trace: "+e.toString());
39                         break;
40                     }
41                 }
42             });
43     }
```

```
43
44     readMessage.start();
45     String line = "";
46     while (!(line.equals("Over"))) {
47         try {
48
49             // read the message sent to this client
50             line = input.readLine();
51             out.writeUTF(line);
52         } catch (IOException e) {
53             System.out.println("Unknown Error Occured while sending Message! Trace: "
+e.toString());
54             break;
55         }
56     }
57     readMessage.stop();
58     try{
59         input.close();
60         out.close();
61         socket.close();
62     }catch(IOException i){
63         System.out.println(i);
64     }
65
66 }
67
68 public static void main(String args[]){
69     Client client = new Client("127.0.0.1", 5000);
70 }
71 }
```

## Server:

```
1
2 import java.io.*;
3 import java.net.*;
4
5 public class Server{
6
7     private Socket socket = null;
8     private ServerSocket server = null;
9     private DataInputStream in = null;
10    private DataOutputStream out = null;
11    private DataInputStream input = null;
12    public Server(int port){
13
14        try{
15
16            server = new ServerSocket(port);
17
18            System.out.println("Server started");
19            while(true){
20                System.out.println("Waiting for a client ...");
21                socket = server.accept();
22                System.out.println("Client accepted");
23
24                in = new DataInputStream(
25                    new BufferedInputStream(socket.getInputStream()));
26                out = new DataOutputStream(socket.getOutputStream());
27                input = new DataInputStream(System.in);
28
29                Thread writeMessage = new Thread(new Runnable() {
30
31                    @Override
32                    public void run() {
33                        while (true){
34                            try{
35                                String inputString = input.readLine();
36                                out.writeUTF(inputString);
37                            }catch(IOException i){
38                                System.out.println("Lost connection with socket!");
39                                break;
40                            }
41                        }
42                    }
43                });
44                writeMessage.start();
45            }
46        }
47    }
48 }
```

```

46         String line = "";
47         while (!line.equals("Over")) {
48             try {
49                 // read the message sent to this client
50                 line = in.readUTF();
51                 System.out.println("Client: " + line);
52             } catch (IOException e) {
53                 System.out.println(
54                     "Unknown Error Occured while reading Message! Trace: " + e.toString());
55                 break;
56             }
57             writeMessage.stop();
58             System.out.println("Closing connection");
59             socket.close();
60             in.close();
61         }
62     } catch (IOException i) {
63         System.out.println(i);
64     }
65 }
66 public static void main(String args[]) {
67     Server server = new Server(5000);
68 }
69 }

```

Note: The Server code runs infinitely long unless stopped otherwise and receives client requests even if the client wishes to leave and rejoin again.

## Output:

```

^Ckrhero@hellblazer: /mnt/0FB812900FB81290/BTech/Assignments/3rd_Yea
ear/CN/Assignment3$ java Server.java
Note: Server.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Server started
Waiting for a client ...
Client accepted
Client: Hey
Hi there buddy
Client: So, what's up?
Nothing, Just implemented a Server where you can leave and rejoin
.
Client: Thats great...
Client: So, what if I leave and rejoin?
Go ahead...
Client: Over
Closing connection
Waiting for a client ...
Client accepted
Client: Hey I am back
And I am still running LOL
Client: Okay
Yeah
█

krhero@hellblazer: /mnt/0FB812900FB81290/BTech/Assignments/3rd_Yea
r/CN/Assignment3$ java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Hey
Server: Hi there buddy
So, what's up?
Server: Nothing, Just implemented a Server where you can leave and rejoin.
Thats great...
So, what if I leave and rejoin?
Server: Go ahead...
Over
krhero@hellblazer: /mnt/0FB812900FB81290/BTech/Assignments/3rd_Yea
r/CN/Assignment3$ java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Hey I am back
Okay
Server: Yeah
█

```