

EE321 Assignment 1

Time Server Application

Aim of the assignment

The goal of this assignment is to create a basic server-side application that accepts a connection from a client and outputs to the client a string of the current date and time of the server. My goal with this assignment is to learn the basics of running a server-side java application, setting up a connection with a client application, and output data through the connection.

Procedure

The first step was to read the assignment document given to us it explained the basics of how sockets and read and write operations work in java. Using this explanation and example code I began by setting up a server-side socket that simply sent a test message to any client that connected to it.

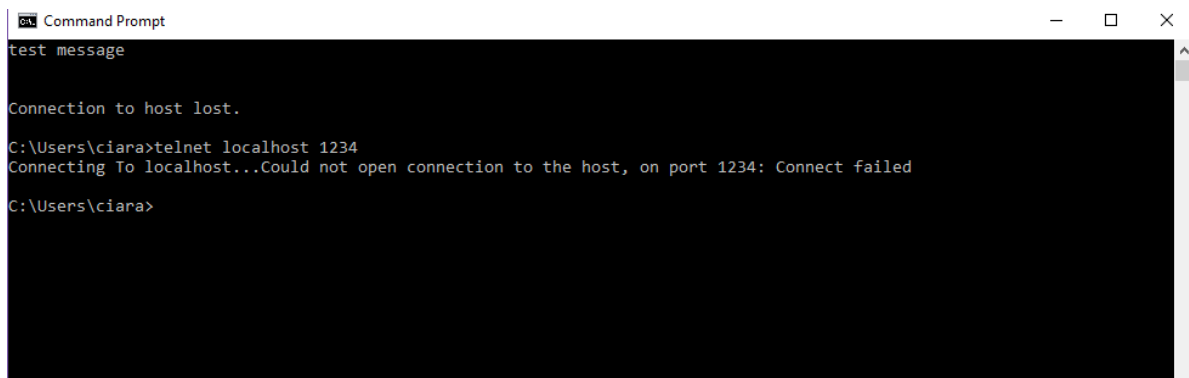
I began by simply constructing a socket at port 1234 and constructing a client socket that would be initialized by my server socket accepting it, and a `PrintWriter` not yet initialized to anything.

```
ServerSocket myServerSocket = new ServerSocket(1234);
PrintWriter out = null;
Socket connectedClientSocket = myServerSocket.accept();
```

I then initialized my `PrintWriter` to my client sockets output stream so anything printed to this would be printed through the output stream to the client. I tested this by printing a test message and then closed the stream.

```
out = new PrintWriter(connectedClientSocket.getOutputStream(), true);
out.println("test message");
out.close();
```

Now this code worked however it would only accept the first connection to the server print the message and then the process would end and no further clients could connect.



```
Command Prompt
test message

Connection to host lost.

C:\Users\ciara>telnet localhost 1234
Connecting To localhost...Could not open connection to the host, on port 1234: Connect failed
C:\Users\ciara>
```

So then I used a simple while(true) loop to keep the code looping, making sure to close the client socket after sending the message in order to allow a new client to connect.

```
ServerSocket myServerSocket = new ServerSocket(1234);
PrintWriter out = null;
Socket connectedClientSocket = null;
while(true) {
    connectedClientSocket = myServerSocket.accept();
    out = new PrintWriter(connectedClientSocket.getOutputStream(), true);
    out.println("test message");
    out.close();
    connectedClientSocket.close();
}
```

With this code the server now accepted a connection, output the test message to the client, disconnected from the client and waited for the next client to try and connect.



And repeating the telnet command gave the same output.

With my server application now capable of output as required I began it was time to implement the code to output the required message, in this case the date and time of the machine. This simply meant creating a string with the date and time and printing it to the client socket output stream. This was done using:

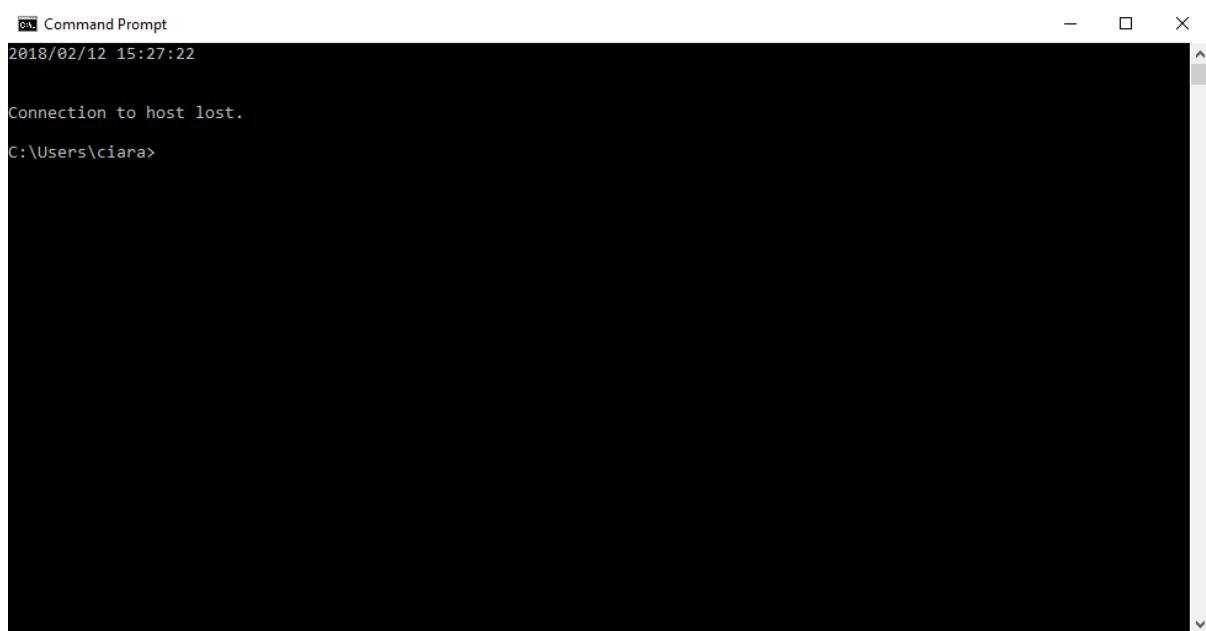
```
String date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(new Date());
out.println(date);
```

This was used in place of the test message.

Results

With this code in place my server now allows a client to connect, prints to the output stream of the client socket the date and time and then closes the connection and waits for another client to try and connect. My server application therefore fits the brief and works correctly.

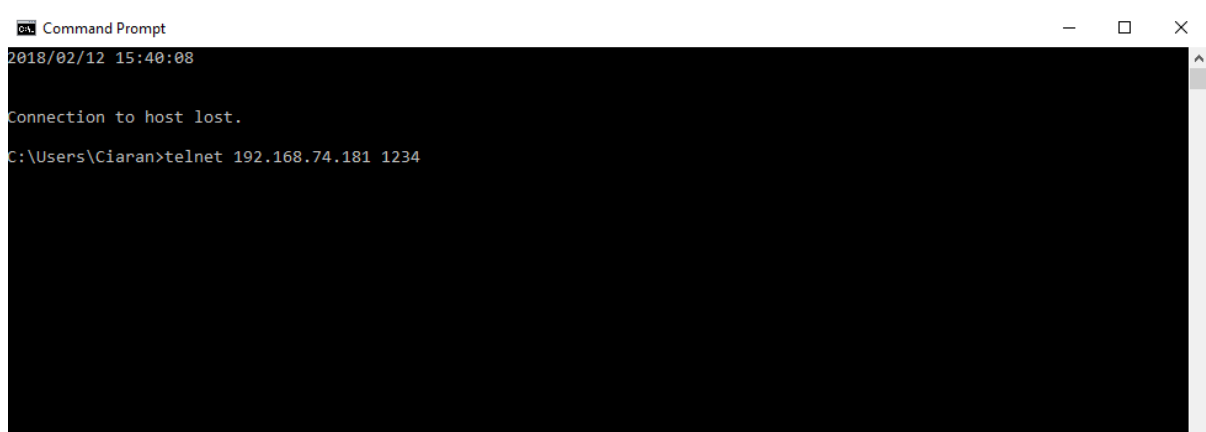
I tested the code by using the command `telnet localhost 1234` and this was the output.



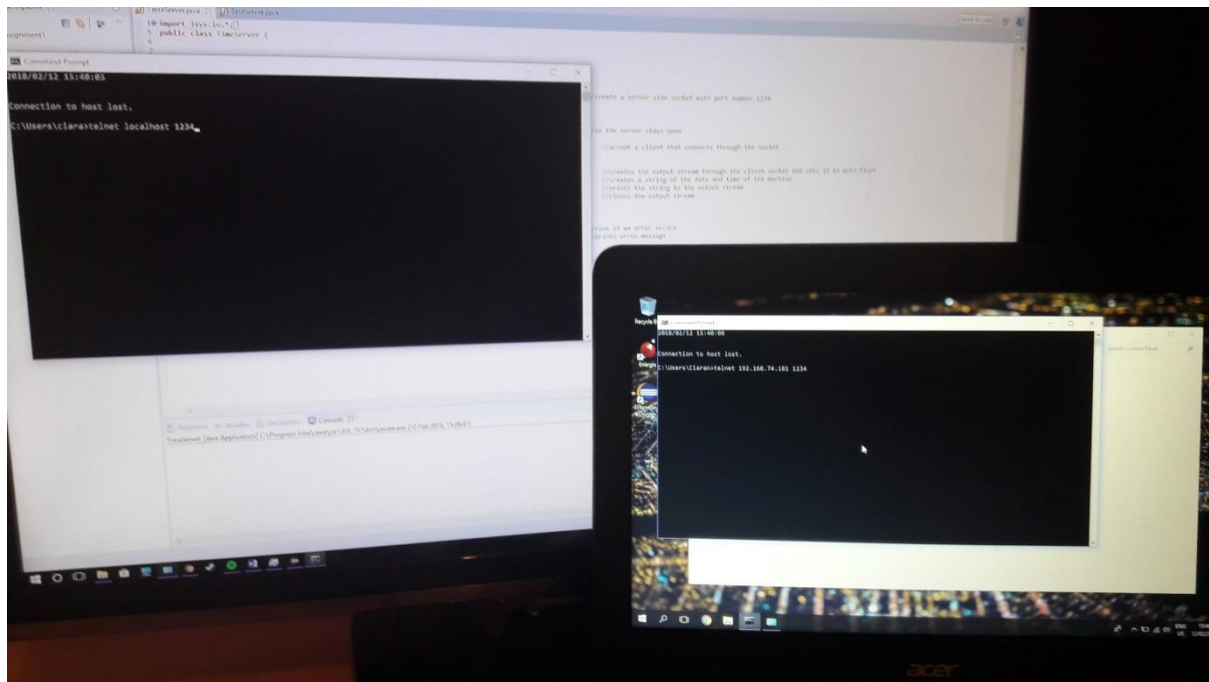
```
Command Prompt
2018/02/12 15:27:22
Connection to host lost.
C:\Users\ciara>
```

The code worked for repeated inputs.

I then tested it from a separate machine connected to the same network using the command `telnet <server_machine_local_IP> 1234` and again the output was correct, and again worked for multiple connections.



```
Command Prompt
2018/02/12 15:40:08
Connection to host lost.
C:\Users\Ciaran>telnet 192.168.74.181 1234
```



Source Code

```
import java.io.*;
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.Date;
public class TimeServer {

    public static void main (String[] args) throws IOException{
        try {
            ServerSocket myServerSocket = new ServerSocket(1234);
                                                    //create a server side socket with port number 1234
            PrintWriter out = null;

            while(true) {
                //so the server stays open

                Socket connectedClientSocket = myServerSocket.accept();
                                                    //accept a client that connects through the socket

                out = new PrintWriter(connectedClientSocket.getOutputStream(), true);
                                                    //creates the output stream through the client socket and sets it to auto flush
                String date = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(new Date());
                                                    //creates a string of the date and time of the machine
                out.println(date); //prints the string to the output stream
                out.close();      //closes the output stream
                connectedClientSocket.close();
            }

        } catch(IOException e) {
                                                    //runs if an error occurs
            System.out.println("Error:" + e.getMessage());
                                                    //prints error message
        }
    }
}
```