CS 4390 Computer Networks

Chat Program Project Write-up

Alex Szeto – acs140530

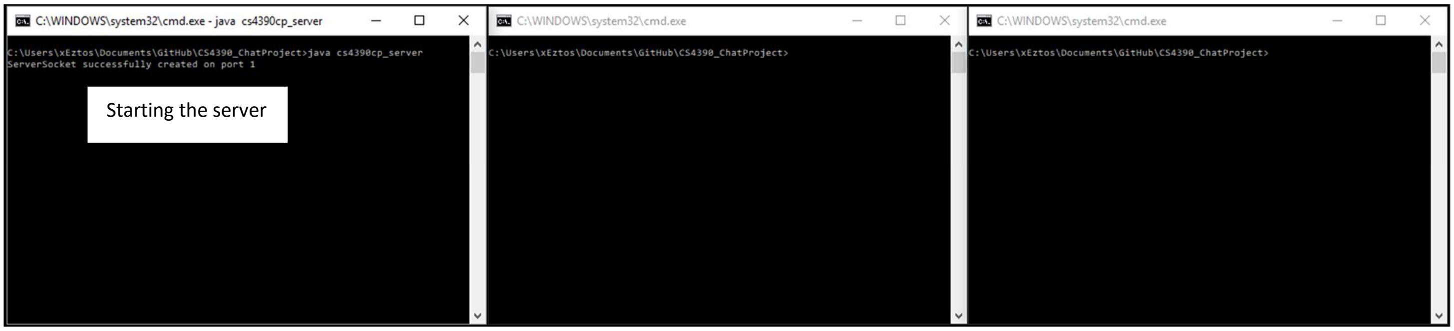
The objective of this project was to design and to program, using socket programming, a simple chat session application that allows for two people to chat with each other. The program that I developed is run on the command console, and allows two people, and theoretically even more people to communicate in a group chat together, with each person's message broadcasted to the other people paired with a unique identifier. The program will also notify the other users if a user disconnects, and the users will also be notified if the server was suddenly disconnected.

An initial hiccup that I ran into was the fact that I did not understand what the term "socket programming" meant, since I have never heard the term before. A quick google of the term revealed that the term merely referred to the idea that programs were talking to using open listening port sockets. I then did much research into the java.net networking API.

Initially, the server accepted only one connection at a time, but that does not allow for two-way communication. The server then used a while loop to continuously accept connection requests, but only one client was able to communicate with the server at a time. The threaded clientComm was implemented to allow multiple concurrent connections to send messages to the server and the server will receive each one of them. An array list was put outside in the main class populated with the printwriter outputs for each individual client, however, because the individual clientComm was not able to communicate with all the other clientComm threads.

After I got a general framework hammered out, I very quickly realized that the client-side program was not able to both listen to the server messages and listen to the user console messages at the same time. Luckily, the projects in CS 4348 Operating Systems helped me realize that if the monitoring of both server-side messages and client-side messages were handled on separate threads within the same program, they would be able to run concurrently. That lead to the "extend" of the thread in the server-side monitoring client. The Java Scanner package already runs in its own thread, meaning that I did not have to implement the scanner as its own thread.

The following series of screen captures represent a sample program execution:

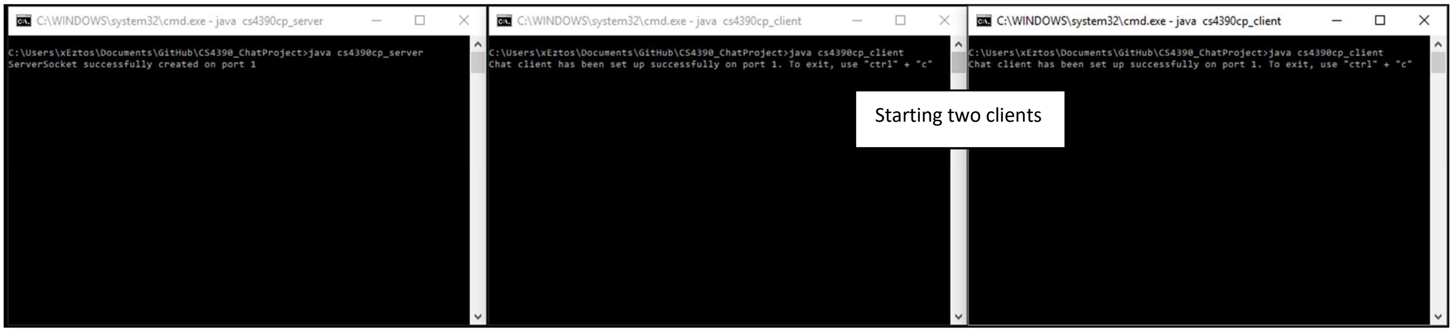

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

Starting the server

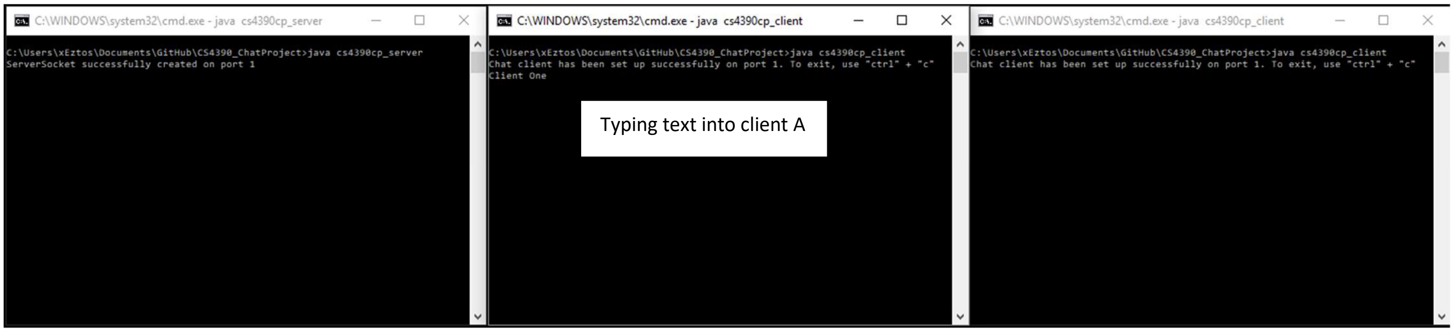

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
```

Starting two clients

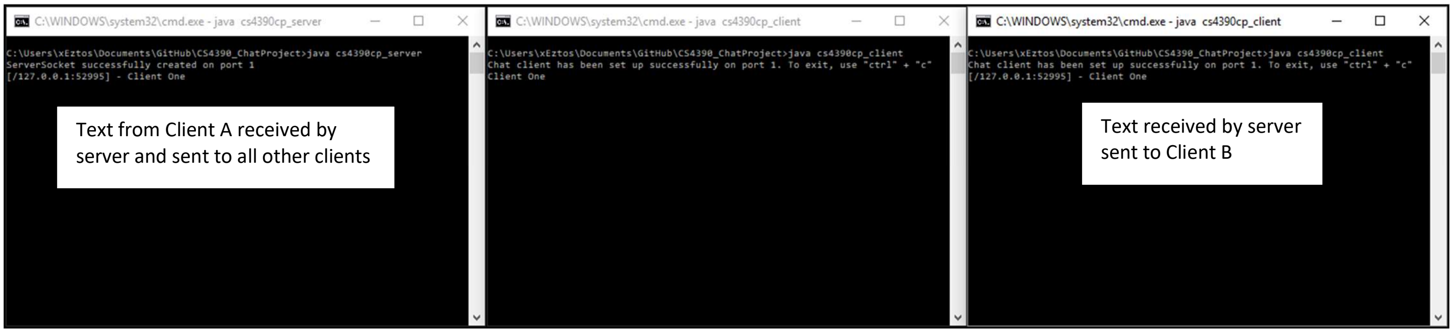

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
```

Typing text into client A


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

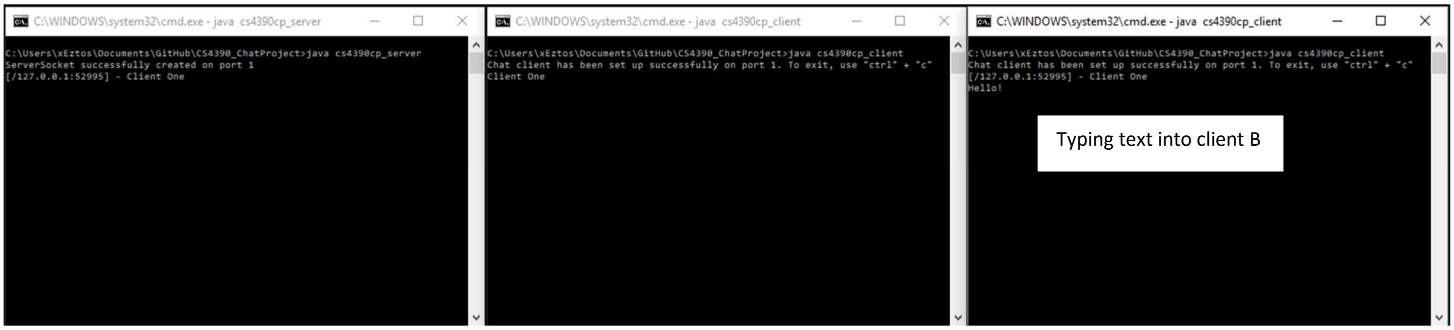
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:52995] - Client One
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:52995] - Client One
```

Text from Client A received by server and sent to all other clients

Text received by server sent to Client B

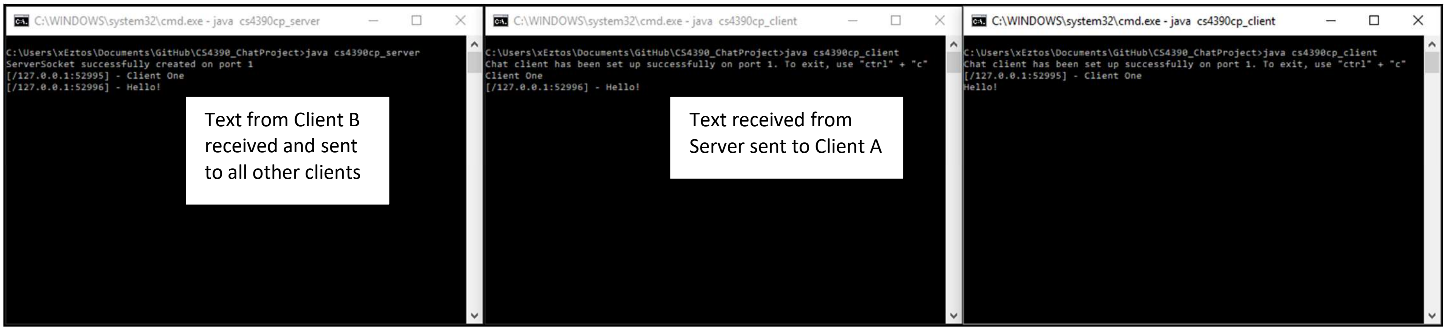

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:52995] - Client One
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:52995] - Client One
Hello!
```

Typing text into client B


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

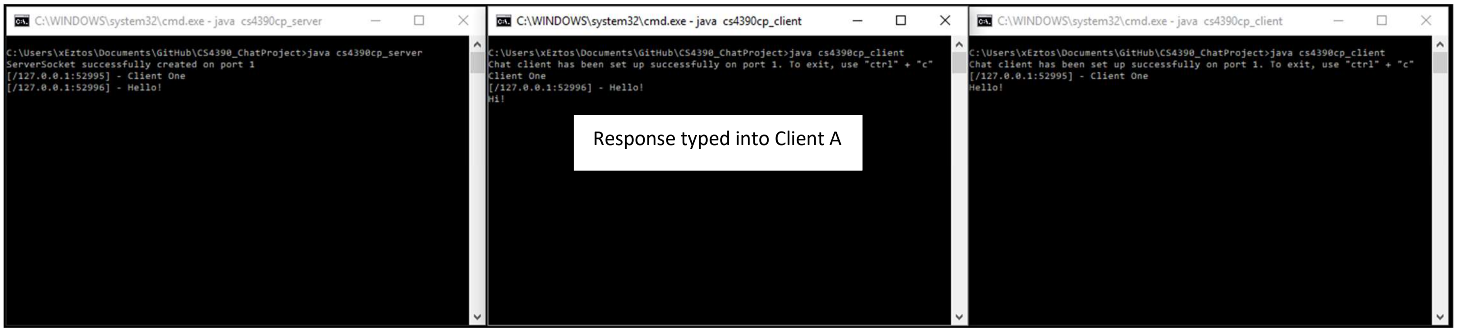
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:52995] - Client One
[/127.0.0.1:52996] - Hello!
```

Text from Client B received and sent to all other clients

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:52996] - Hello!
```

Text received from Server sent to Client A

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:52995] - Client One
Hello!
```

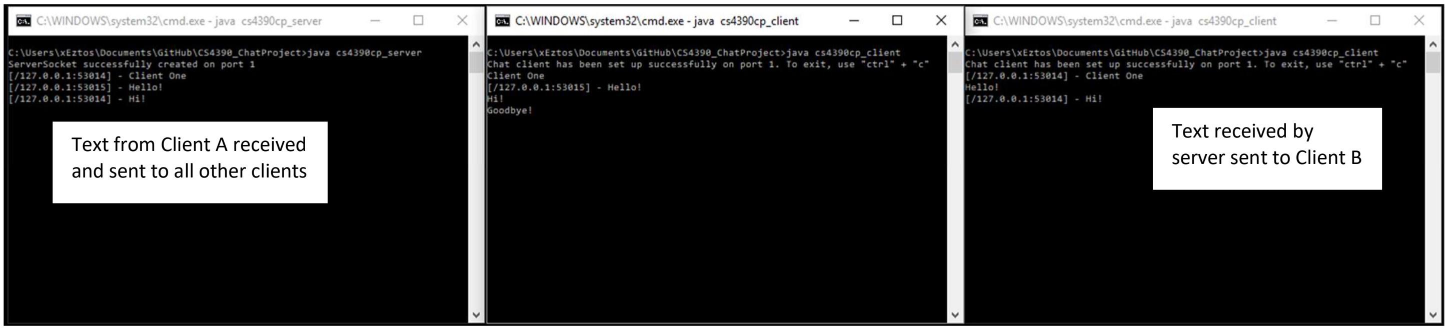

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:52995] - Client One
[/127.0.0.1:52996] - Hello!
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:52996] - Hello!
Hi!
```

Response typed into Client A

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:52995] - Client One
Hello!
```


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

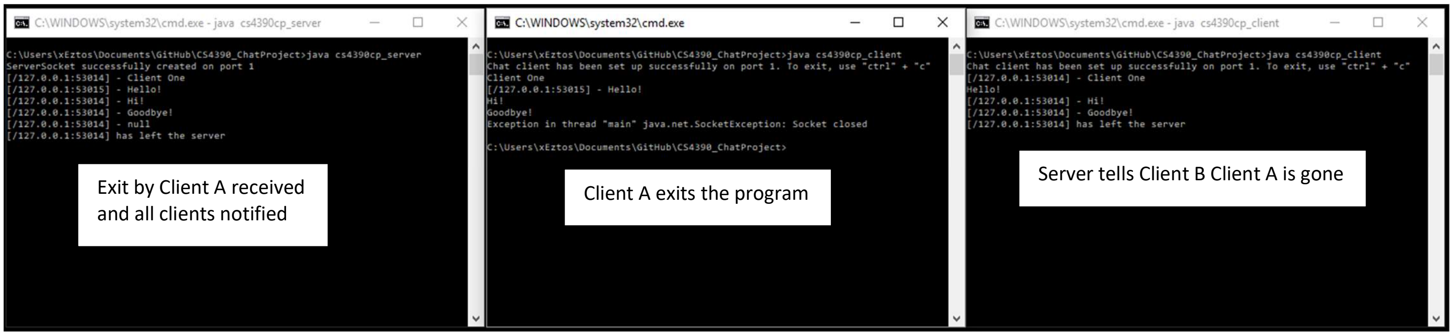
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
```

Text from Client A received and sent to all other clients

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
```

Text received by server sent to Client B


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] - null
[/127.0.0.1:53014] has left the server
```

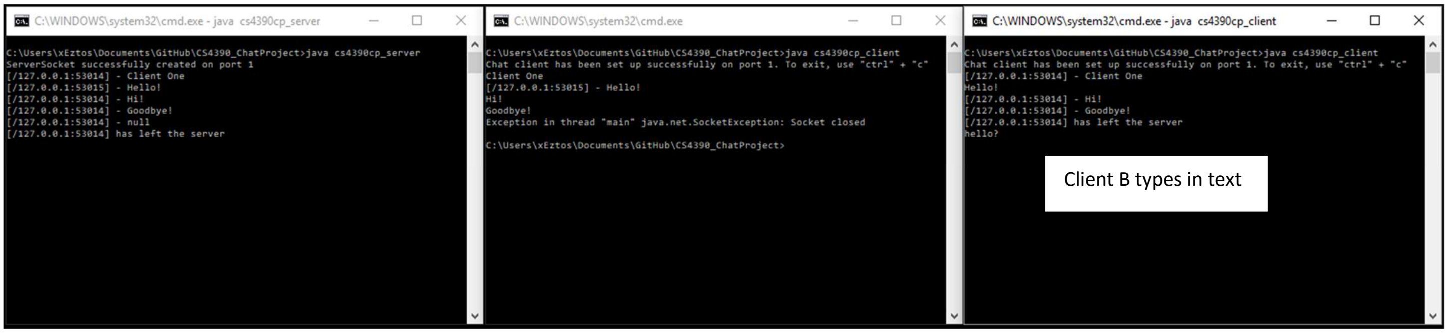
Exit by Client A received and all clients notified

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
Exception in thread "main" java.net.SocketException: Socket closed

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

Client A exits the program

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] has left the server
```

Server tells Client B Client A is gone

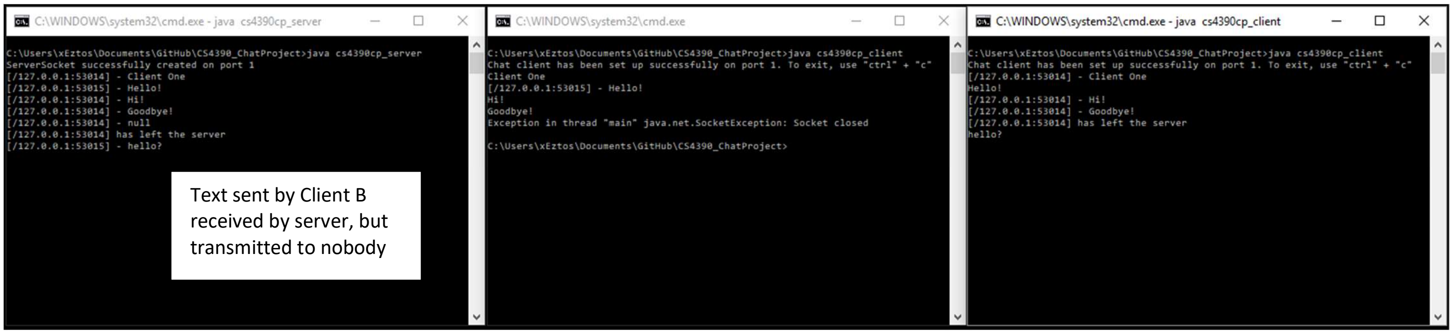

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] - null
[/127.0.0.1:53014] has left the server
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
Exception in thread "main" java.net.SocketException: Socket closed

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] has left the server
hello?
```

Client B types in text

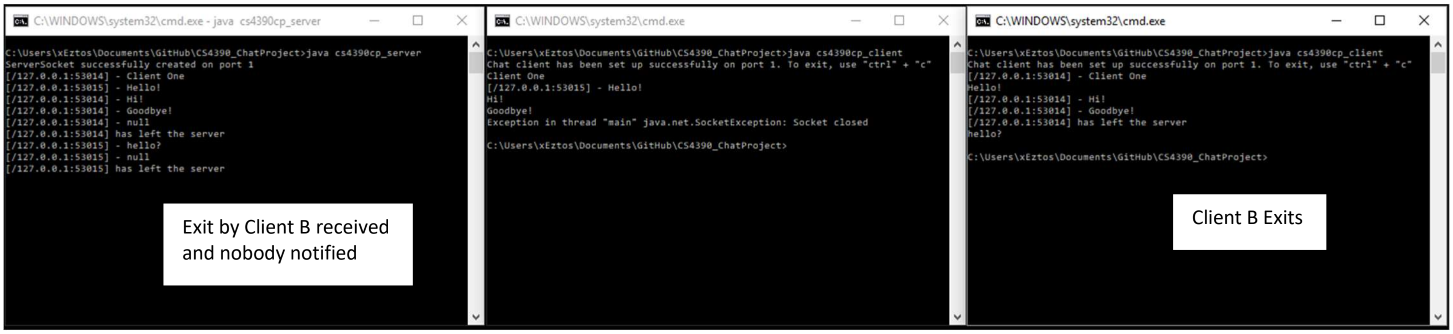

C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] - null
[/127.0.0.1:53014] has left the server
[/127.0.0.1:53015] - hello?
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
Exception in thread "main" java.net.SocketException: Socket closed

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] has left the server
hello?
```

Text sent by Client B received by server, but transmitted to nobody


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe

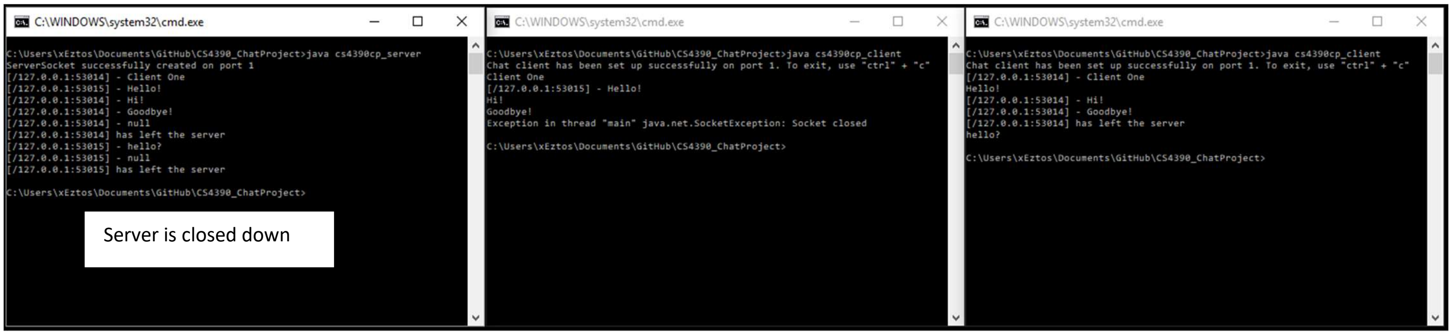
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] - null
[/127.0.0.1:53014] has left the server
[/127.0.0.1:53015] - hello?
[/127.0.0.1:53015] - null
[/127.0.0.1:53015] has left the server
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
Exception in thread "main" java.net.SocketException: Socket closed

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] has left the server
hello?

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

Exit by Client B received and nobody notified

Client B Exits


C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe

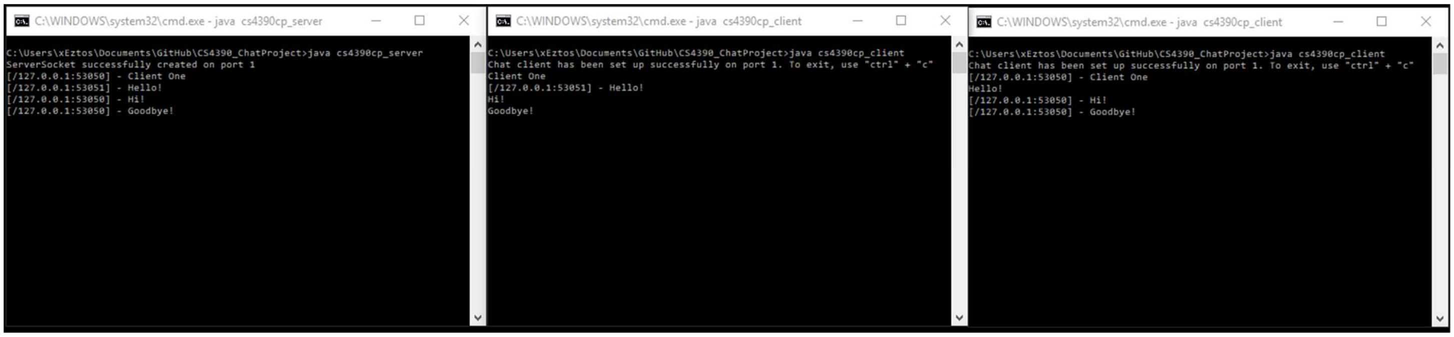
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53014] - Client One
[/127.0.0.1:53015] - Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] - null
[/127.0.0.1:53014] has left the server
[/127.0.0.1:53015] - hello?
[/127.0.0.1:53015] - null
[/127.0.0.1:53015] has left the server

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53015] - Hello!
Hi!
Goodbye!
Exception in thread "main" java.net.SocketException: Socket closed

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53014] - Client One
Hello!
[/127.0.0.1:53014] - Hi!
[/127.0.0.1:53014] - Goodbye!
[/127.0.0.1:53014] has left the server
hello?

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

Server is closed down

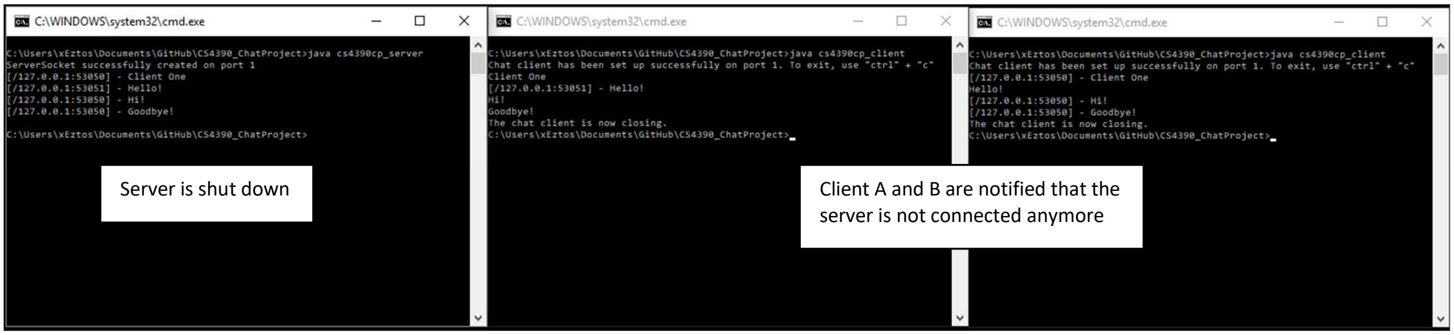
These Two images represent the server closing while there are clients connected:


C:\WINDOWS\system32\cmd.exe - java cs4390cp_server
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client
C:\WINDOWS\system32\cmd.exe - java cs4390cp_client

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53050] - Client One
[/127.0.0.1:53051] - Hello!
[/127.0.0.1:53050] - Hi!
[/127.0.0.1:53050] - Goodbye!
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53051] - Hello!
Hi!
Goodbye!
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53050] - Client One
Hello!
[/127.0.0.1:53050] - Hi!
[/127.0.0.1:53050] - Goodbye!
```


C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\cmd.exe

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_server
ServerSocket successfully created on port 1
[/127.0.0.1:53050] - Client One
[/127.0.0.1:53051] - Hello!
[/127.0.0.1:53050] - Hi!
[/127.0.0.1:53050] - Goodbye!

C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

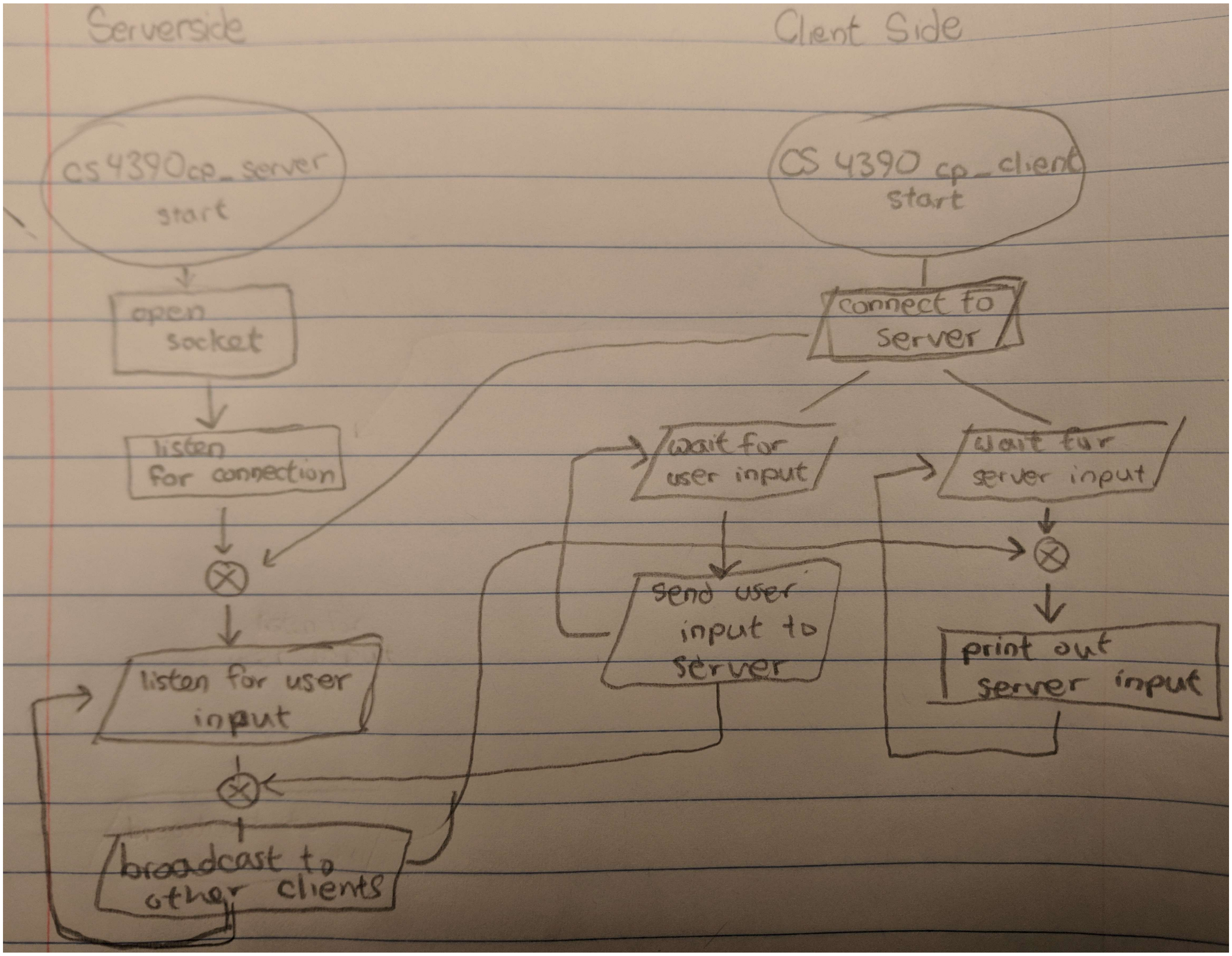
```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
Client One
[/127.0.0.1:53051] - Hello!
Hi!
Goodbye!
The chat client is now closing.
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

```
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>java cs4390cp_client
Chat client has been set up successfully on port 1. To exit, use "ctrl" + "c"
[/127.0.0.1:53050] - Client One
Hello!
[/127.0.0.1:53050] - Hi!
[/127.0.0.1:53050] - Goodbye!
The chat client is now closing.
C:\Users\xEztos\Documents\GitHub\CS4390_ChatProject>
```

Server is shut down

Client A and B are notified that the server is not connected anymore

Flowchart:


Serverside
Client Side
CS 4390cp_server start
open socket
listen for connection
listen for user input
broadcast to other clients
CS 4390 cp_client start
connect to server
wait for user input
send user input to server
wait for server input
print out server input

Server Code:

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

public class cs4390cp_server{
    static int port = 1;
    static ArrayList<PrintWriter> pw = new ArrayList<PrintWriter>();

    /**
     * Main metod of the server. This method sets up a socket, and attempts to continuously accept any incomming connection requests by a client
     * @param args not used
     */
    public static void main(String[] args) throws IOException{
        ServerSocket socket = null;
        try{
            socket = new ServerSocket(port);
            System.out.printf("ServerSocket successfully created on port %d%n", port);
        } catch (IOException e){
            System.out.println("ERROR FLAG 5");
            System.exit(-1);
        }

        try{
            while(true){
                new ClientComm(socket.accept()).start();
            }
        } finally{
            try{
                socket.close();
            } catch (Exception e){
                System.out.println("ERROR: FLAG 0");
                System.exit(-1);
    }}}

    /**
     * Subclass of the client communication that is paired with a single connected chat client.
     */
    private static class ClientComm extends Thread{
        Socket socket;
        BufferedReader input;
        PrintWriter output;

        /**
         * Constructor for a client communicator to a specified socket
         * @param socket the socket the cilent will be communicating from
         */
        ClientComm(Socket socket){
            this.socket = socket;
        }

        /**
         * The run method invoked by the start() of the extended thread class. This method
         * will read what a client is communicating to this server program and tell all the
         * other connected clients what this specific clientComm is associated with is saying
         */
        public void run(){
            try{
                input = new BufferedReader(new InputStreamReader(socket.getInputStream()));
                output = new PrintWriter(socket.getOutputStream(), true);

                pw.add(output);          // adds a client PrintWriter into an arrayList of printwriters

                while(true){
                    String temp = input.readLine();
                    String ret = String.format("[%s:%d] - %s",socket.getLocalAddress().toString(),socket.getPort(),temp);
                    System.out.println(ret);

                    if(temp == null){
                        String message = String.format("[%s:%d] has left the server", socket.getLocalAddress().toString(), socket.getPort());
                        System.out.println(message);
                        broadcast(message);
                        return;
                    } else if(pw.size() == 1){
                        output.printf("SERVER - There is nobody else on the server.");
                    }else{
                        for(int i = 0; i < pw.size(); i++){

                            PrintWriter tempW = pw.get(i);
                            if(tempW != output){
                                    tempW.println(ret);
                }}}}
            } catch(IOException e){     // client forcefully (ctrl + c) disconnects
                System.out.println("ERROR: FLAG 1");
                return;
            } finally{
                pw.remove(output);
                try{
                    socket.close();
                } catch( IOException e){
                    System.out.println("ERROR: FLAG 4");
        }}}

        /**
         * A method used by the server to broadcast a message to all the users connected to this server
         * @param s the message in the form of a string to broadcast
         */
        public void broadcast(String s){
            for(int i = 0; i < pw.size(); i++){
                PrintWriter tempW = pw.get(i);
                tempW.println(s);
    }}}}
```

Client Code:

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

/**
 * Main class for the chat client. Enables a user to chat with fellow users after connecting to a central server
 */
public class cs4390cp_client{
    static int port = 1;
    static Socket socket;
    static PrintWriter toServer;
    static BufferedReader fromServer;

/**
 * Main method. Sets up a socket, probes for a valid server, and communicates with the server.
 * @args not used
 */
    public static void main(String[] args) throws IOException{
        try{
            try{
                socket = new Socket("localhost", port);
            } catch (UnknownHostException e){
                System.out.println("ERROR: cs4390cp_client FLAG 0");
                System.exit(-1);
            } catch (IOException e){
                System.out.println("ERROR: cs4390cp_client FLAG 1");
                System.exit(-1);
            }

            toServer = new PrintWriter(socket.getOutputStream(), true);
            new serverRecieve().start();

            Scanner sc = new Scanner(System.in);

            System.out.println("Chat client has been set up successfully on port " + port + ". To exit, use \"ctrl\" + \"c\"");

            while(true){
                String temp = sc.nextLine();
                toServer.println(temp);
        }} finally{
            socket.close();
    }}

/**
 * Subclass that enables the chat client to recieve messages from the chat server on a seperate thread, enabling for concurrent user and server inputs.
 */
    private static class serverRecieve extends Thread{
/**
 * run method run by the thread class upon calling of the extended start method.
 */
        public void run(){
            try{
                fromServer = new BufferedReader(new InputStreamReader(socket.getInputStream()));
                while(true){
                    String temp = fromServer.readLine();
                    if(temp == null){
                        try{
                            wait();
                        } catch (InterruptedException e){
                            System.out.println("ERROR: serverSend FLAG 1");
                    }} else{
                        System.out.println(temp);
            }}} catch (IOException e){
                System.out.printf("The server has been disconnected. The client is now closing.");
                System.exit(-1);
}}}}
```