# King Abdulaziz University

## Faculty of Computing and Information Technology

## Department of Computer Science

## CPCS-371

| Name | Id |
|---|---|
| Mansour Aldhahri (Leader) | 2036538 |
| Emad Fawaz Alsulami | 2036759 |
| Mouhana Almouhana | 2037213 |
| Waleed Albishri<br>Ebrahim Mansour Sharka | 2037558<br>2036339 |

# Contents

## Introduction

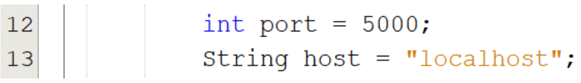
Client and Server is one of the well-known network relationships that partition the workload between the resource (Server) and request sender (Client). They usually communicate over a computer network where a bunch of computers acts as servers and the rest as clients. Also, for instance, the server and client can remain in the same system, and you can simply say that the server is a server program that provides resources and services to the client. On the other hand, the client is a service requester who doesn't provide any resources but requests services from the server. However, in this Project we will create two programs one is running as a server and the other as a client.

## Project Description

The main goal of the server is to calculate the number of character repetitions in a specific text entered by the client. The server will be using TCP as the transport protocol. When the client requests a connection, the server will accept and wait for the client to send the character and text. The server then will respond to the client with the number of character repetitions. Subsequently, the server will be asked for [Y/N] process reputation. The server is designed to keep the connection [On] until the client enters [N].

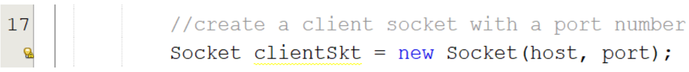
## Client Class

the client-server has two main 2 variables. The first one is the port number and it's an int. The other one is the host name and it's a string.



```
int port = 5000;
String host = "localhost";
```

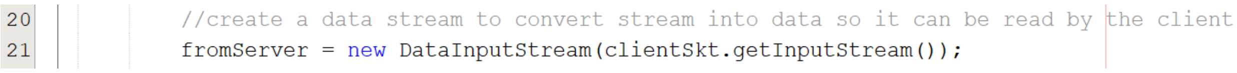
we create a client socket with the host name and port number.



```
//create a client socket with a port number
Socket clientSkt = new Socket(host, port);
```

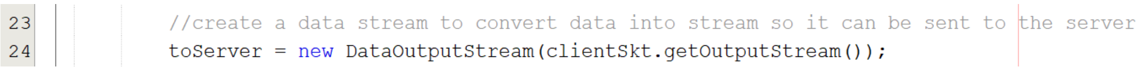
we create 2 data streams to convert stream to data

- the purpose of the first one that can be read by the client.



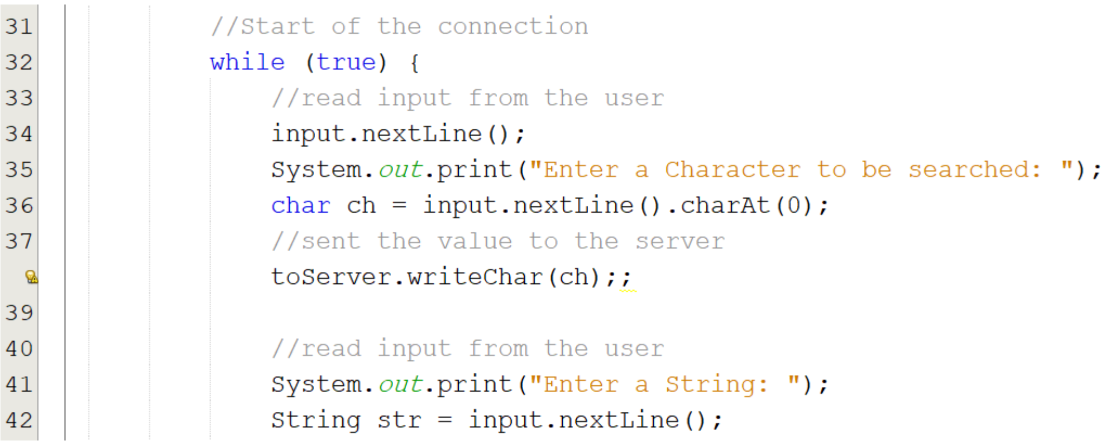
```
//create a data stream to convert stream into data so it can be read by the client
fromServer = new DataInputStream(clientSkt.getInputStream());
```

- the purpose of the second one that can be sent to the server.



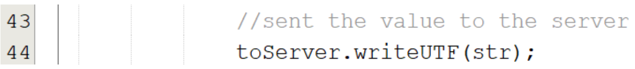
```
//create a data stream to convert data into stream so it can be sent to the server
toServer = new DataOutputStream(clientSkt.getOutputStream());
```

start the connection and read the input from the user



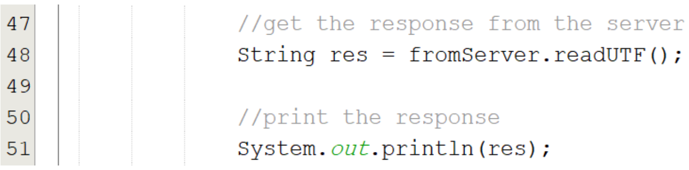
```
//Start of the connection
while (true) {
    //read input from the user
    input.nextLine();
    System.out.print("Enter a Character to be searched: ");
    char ch = input.nextLine().charAt(0);
    //sent the value to the server
    toServer.writeChar(ch);;

    //read input from the user
    System.out.print("Enter a String: ");
    String str = input.nextLine();
```

send the value to the server



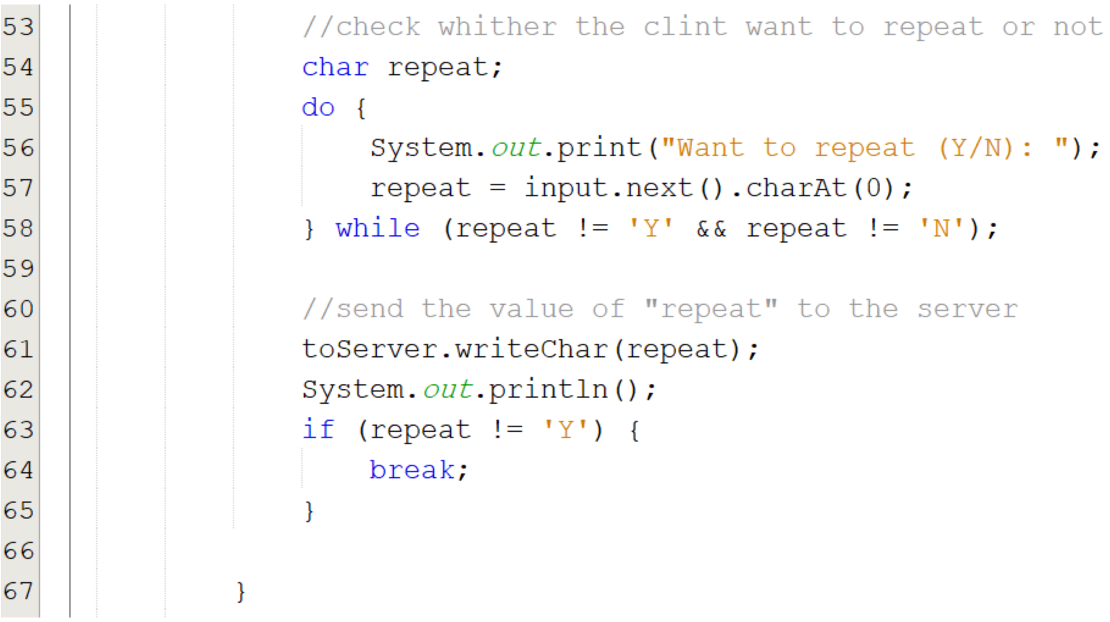
```
//sent the value to the server
toServer.writeUTF(str);
```

get the response from the server and displayed it to the user.



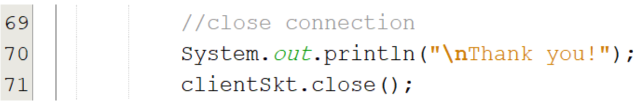
```
//get the response from the server
String res = fromServer.readUTF();

//print the response
System.out.println(res);
```

ask the if he want to repeat or not and send it to the server.



```
        //check whither the clint want to repeat or not
        char repeat;
        do {
            System.out.print("Want to repeat (Y/N): ");
            repeat = input.next().charAt(0);
        } while (repeat != 'Y' && repeat != 'N');

        //send the value of "repeat" to the server
        toServer.writeChar(repeat);
        System.out.println();
        if (repeat != 'Y') {
            break;
        }

    }
```

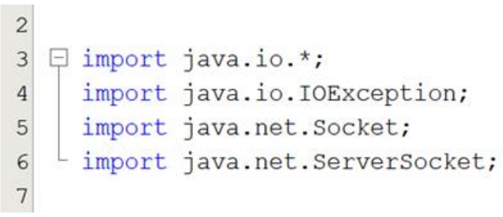
close the connection after you finished.



```
//close connection
System.out.println("\nThank you!");
clientSkt.close();
```

## Server Class

First, import libraries such as the socket library and the server socket library. Start using it in the code and make a server.

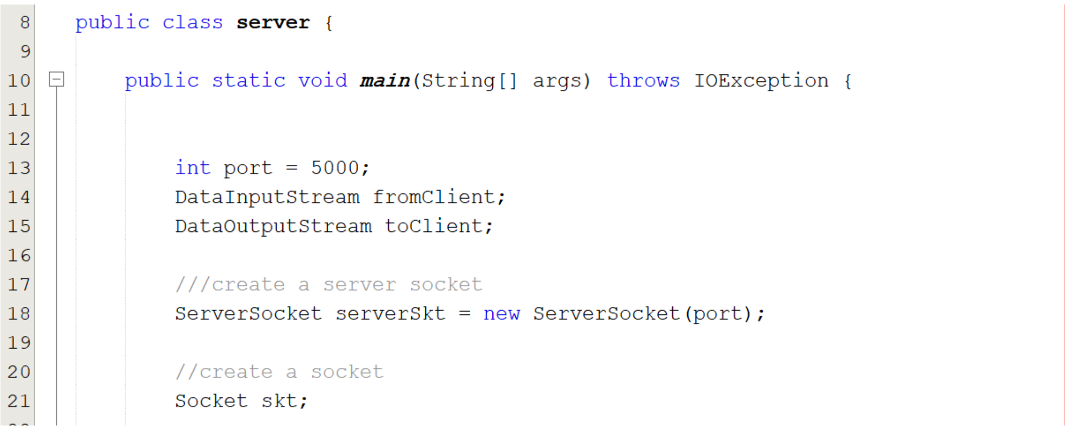


```

import java.io.*;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;

```

In this part, we are working on choosing the port number, which is 5000, and storing it in the variable port.

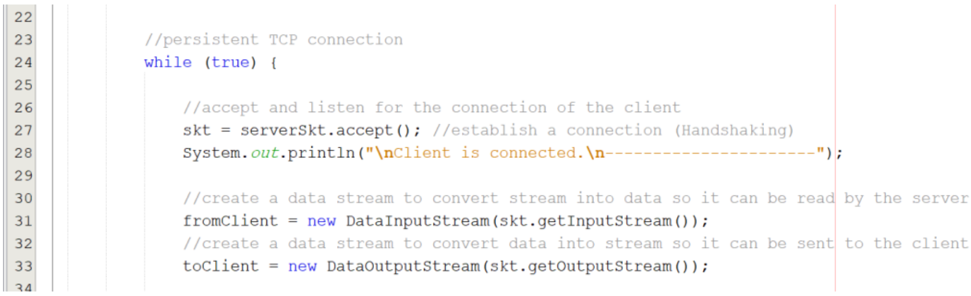
We define an Object of type DataInputStream to store what will come to us from the client and it is called fromclient. We also define an Object of type DataOutputStream to send to the client and it is called toClient.

In line 18 we will create a server socket with port number 5000 to receive any call that comes to us to port. In line 21 we will create a socket for TCP handshake.



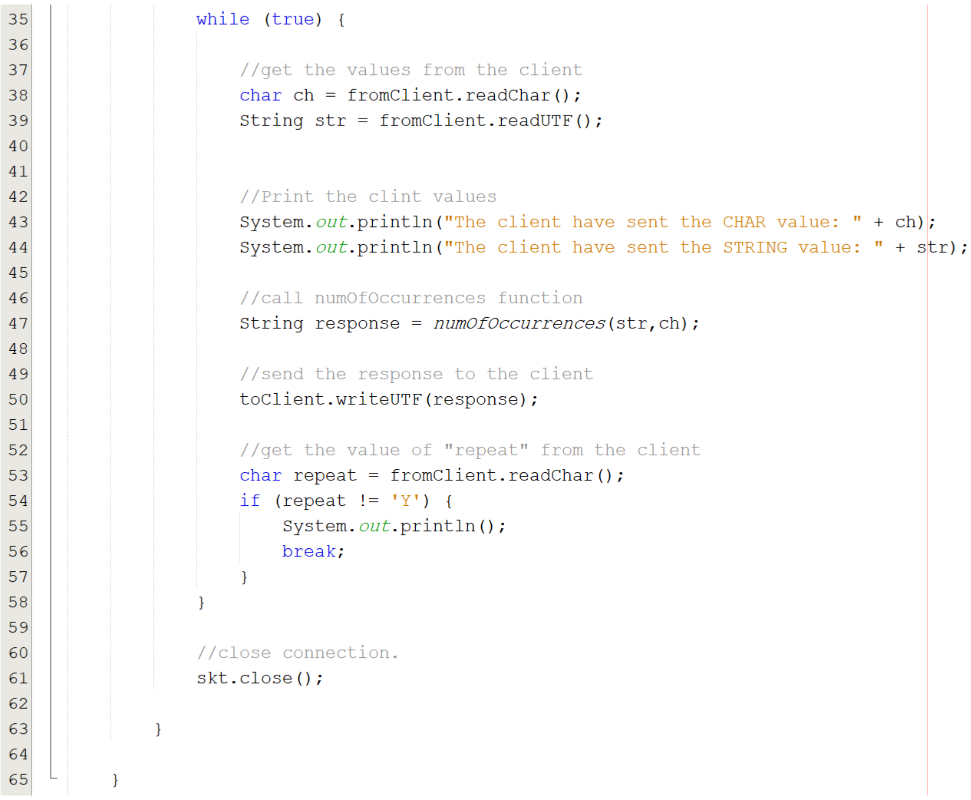
```
 8  public class server {
 9
10      public static void main(String[] args) throws IOException {
11
12
13          int port = 5000;
14          DataInputStream fromClient;
15          DataOutputStream toClient;
16
17          ///create a server socket
18          ServerSocket serverSkt = new ServerSocket(port);
19
20          //create a socket
21          Socket skt;
```

The server is now running as a TCP persistent connection. This means that the connection will be open after receiving the connection request, and the client can send several requests in the same connection. Now the server is waiting for any connection to be made from Client. When a connection occurs, the Client is connected prints, then an Object fromClient is created to take the Input from the client. An Object toClient will be created to send the Output to the Client.



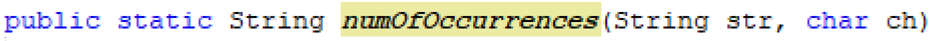
```java

        //persistent TCP connection
        while (true) {

            //accept and listen for the connection of the client
            skt = serverSkt.accept(); //establish a connection (Handshaking)
            System.out.println("\nClient is connected.\n----------------------");

            //create a data stream to convert stream into data so it can be read by the server
            fromClient = new DataInputStream(skt.getInputStream());
            //create a data stream to convert data into stream so it can be sent to the client
            toClient = new DataOutputStream(skt.getOutputStream());
```

The character that the client wants to search for is received and stored in a ch variable of type Char. Also, the text to be searched will be received and stored in the str variable of string. After that, the character sent by the client and the text will be printed on the server The server will call the numOfOccurrences method, which counts the number of character occurrences in the text. After it is called, its value is stored in the response variable and sent to the client via Object toClient using the writeUTF method. When the server is finished, it is waiting for a response from the client. Does it want to repeat the process again? When the answer is a letter other than Y, the server will cut off the connection with the client.



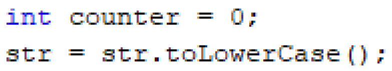
```java
            while (true) {

                //get the values from the client
                char ch = fromClient.readChar();
                String str = fromClient.readUTF();


                //Print the clint values
                System.out.println("The client have sent the CHAR value: " + ch);
                System.out.println("The client have sent the STRING value: " + str);

                //call numOfOccurrences function
                String response = numOfOccurrences(str,ch);

                //send the response to the client
                toClient.writeUTF(response);

                //get the value of "repeat" from the client
                char repeat = fromClient.readChar();
                if (repeat != 'Y') {
                    System.out.println();
                    break;
                }
            }

            //close connection.
            skt.close();

        }

    }
```

## Number Of occurrences Method



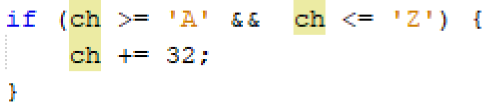
```
public static String numOfOccurrences(String str, char ch)
```

numOfOccurrences function will calculate the number of occurrences of the character in the string, then return a String message says the number of occurrences. It takes two parameters “str”  which is String and  “ch” which is Character.



```
int counter = 0;
str = str.toLowerCase();
```

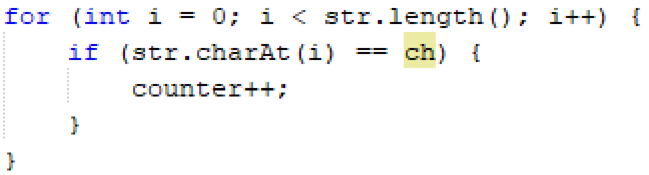
At first, create variable  “counter” of type int to store the number of occurrences, and convert “str” to lowercase.



```
if (ch >= 'A' &&  ch <= 'Z') {
    ch += 32;
}
```

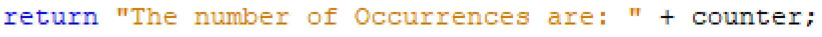
Then, check if the “ch” is uppercase, if so make it lower case by adding 32 (ASCII code)

Now we have the “str” and “ch” both in lowercase



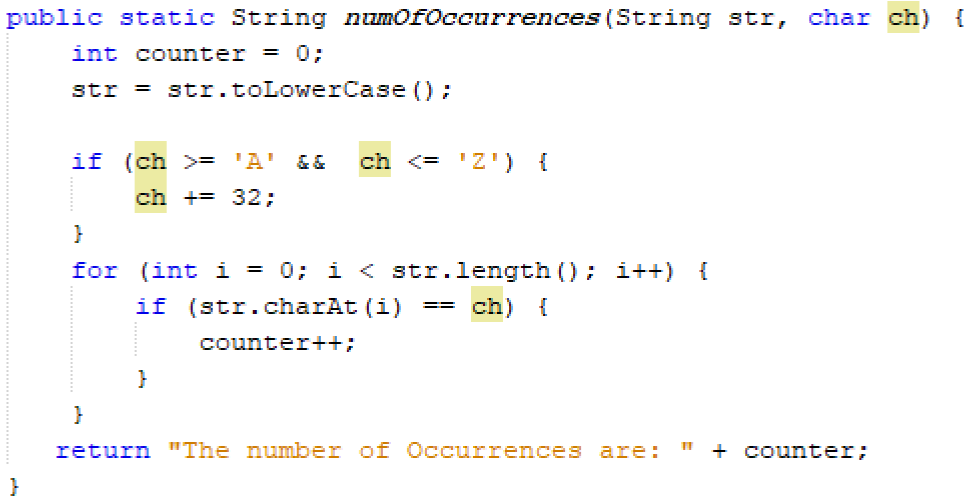
```
for (int i = 0; i < str.length(); i++) {
    if (str.charAt(i) == ch) {
        counter++;
    }
}
```

Then, we will make for loop from 0 to “str” length, and in each iteration, we will check if the “ch” is same as the current character from the “str” , if so we will increment the counter



```
return "The number of Occurrences are: " + counter;
```

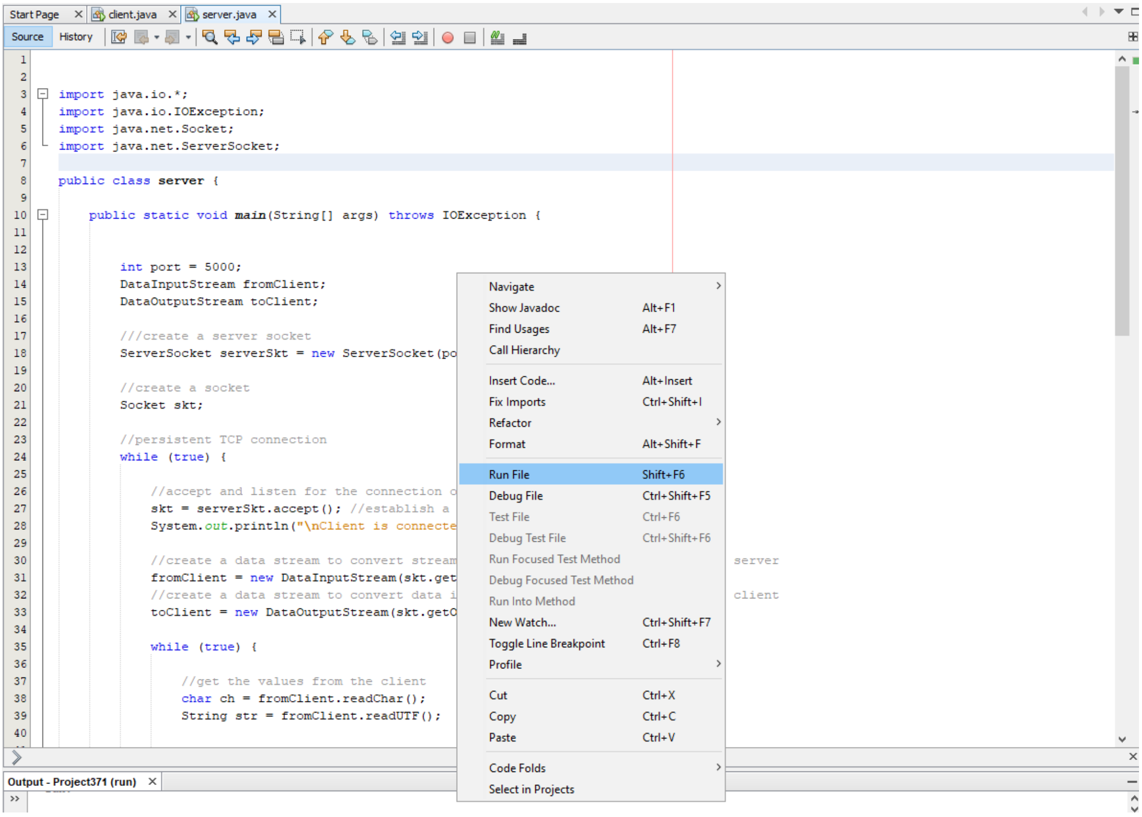
Lastly, return the message ("The number of Occurrences are: " + counter ) as a String



```
public static String numOfOccurrences(String str, char ch) {
    int counter = 0;
    str = str.toLowerCase();

    if (ch >= 'A' &&  ch <= 'Z') {
        ch += 32;
    }
    for (int i = 0; i < str.length(); i++) {
        if (str.charAt(i) == ch) {
            counter++;
        }
    }
   return "The number of Occurrences are: " + counter;
}
```

# Sample run

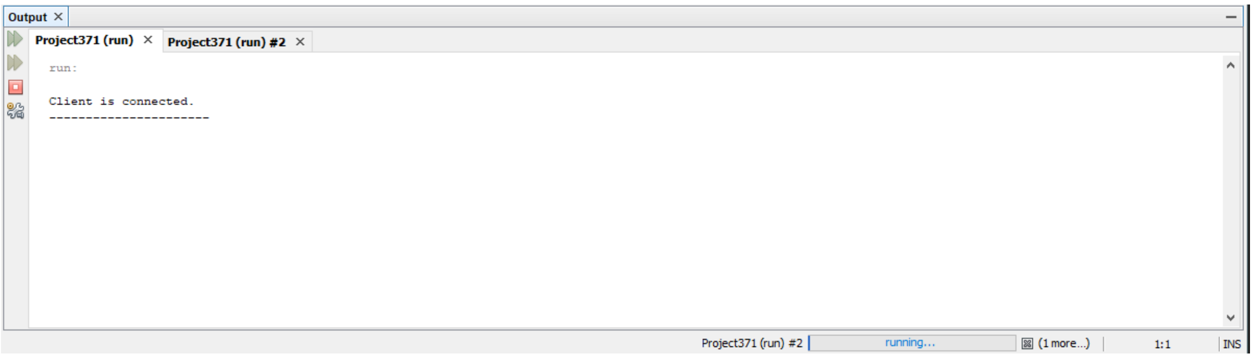
1- We must run the server



```


import java.io.*;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;

public class server {

    public static void main(String[] args) throws IOException {


        int port = 5000;
        DataInputStream fromClient;
        DataOutputStream toClient;

        ///create a server socket
        ServerSocket serverSkt = new ServerSocket(po

        //create a socket
        Socket skt;

        //persistent TCP connection
        while (true) {

            //accept and listen for the connection c
            skt = serverSkt.accept(); //establish a
            System.out.println("\nClient is connecte

            //create a data stream to convert stream                        server
            fromClient = new DataInputStream(skt.get
            //create a data stream to convert data i                        client
            toClient = new DataOutputStream(skt.getO

            while (true) {

                //get the values from the client
                char ch = fromClient.readChar();
                String str = fromClient.readUTF();

```

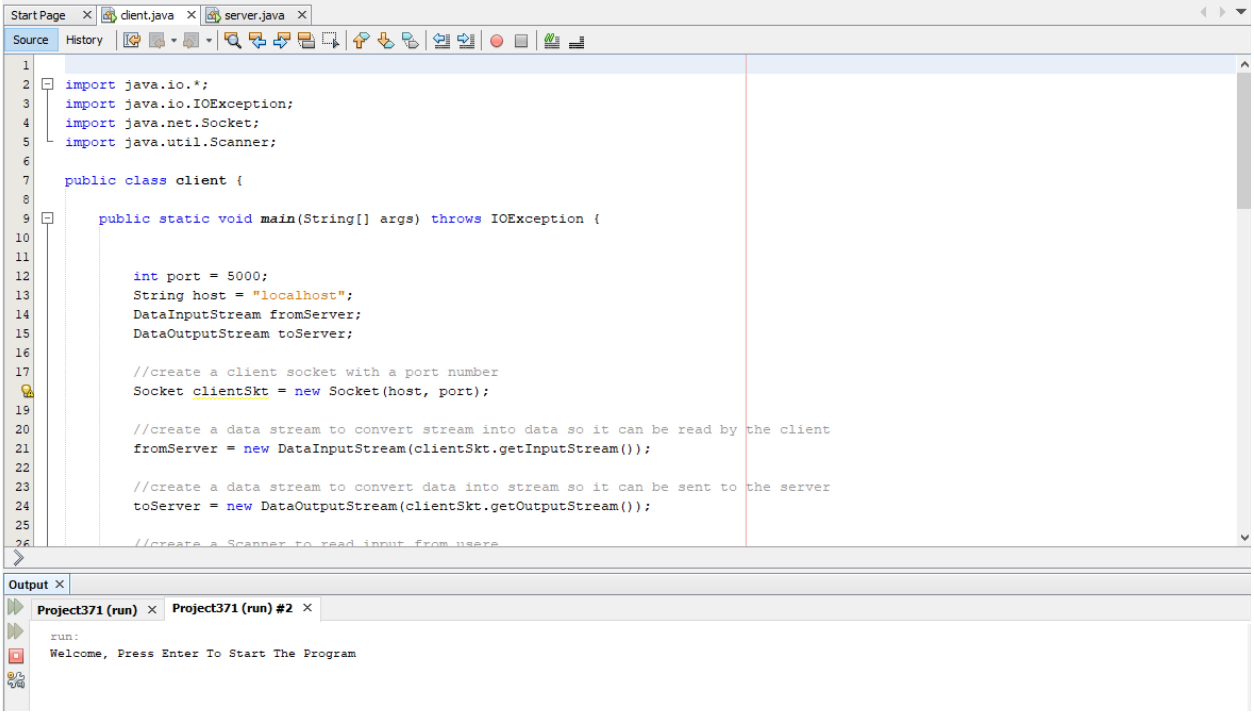
2- Run the Client to establish connection to the server then Press Enter to start the program

Server



```
run:

Client is connected.
----------------------
```

Client

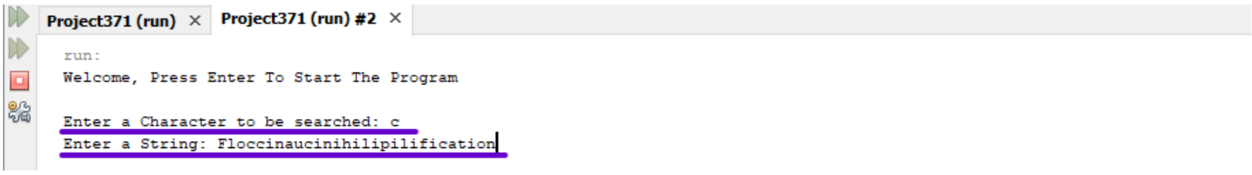


```java

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

public class client {

    public static void main(String[] args) throws IOException {


        int port = 5000;
        String host = "localhost";
        DataInputStream fromServer;
        DataOutputStream toServer;

        //create a client socket with a port number
        Socket clientSkt = new Socket(host, port);

        //create a data stream to convert stream into data so it can be read by the client
        fromServer = new DataInputStream(clientSkt.getInputStream());

        //create a data stream to convert data into stream so it can be sent to the server
        toServer = new DataOutputStream(clientSkt.getOutputStream());

        //create a Scanner to read input from usere
```

```
run:
Welcome, Press Enter To Start The Program
```

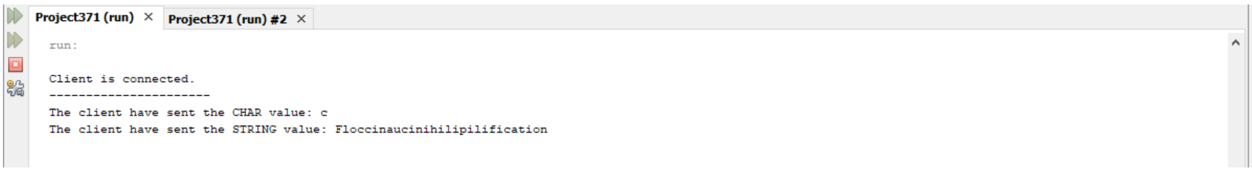
3- Enter a Character to be searched and a String in order to send it to the server

Client



```
run:
Welcome, Press Enter To Start The Program

Enter a Character to be searched: c
Enter a String: Floccinaucinihilipilification
```

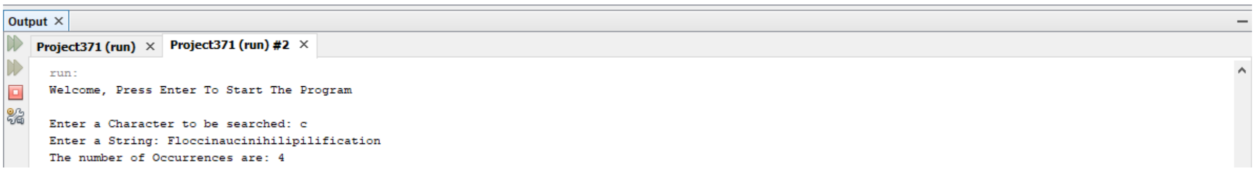
Server



```
run:

Client is connected.
----------------------
The client have sent the CHAR value: c
The client have sent the STRING value: Floccinaucinihilipilification
```

4- The output response from the server

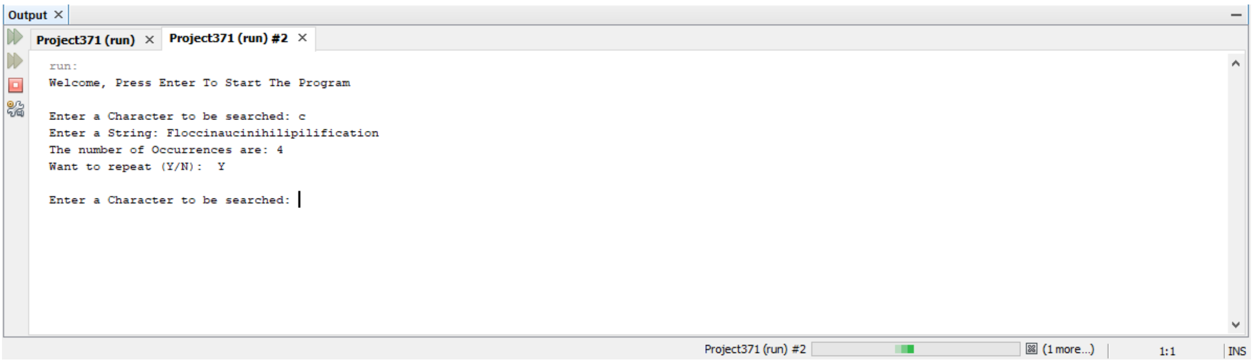
Client



```
run:
Welcome, Press Enter To Start The Program

Enter a Character to be searched: c
Enter a String: Floccinaucinihilipilification
The number of Occurrences are: 4
```

5- Enter ”Y” for process repetition or “N” to disconnect from the server

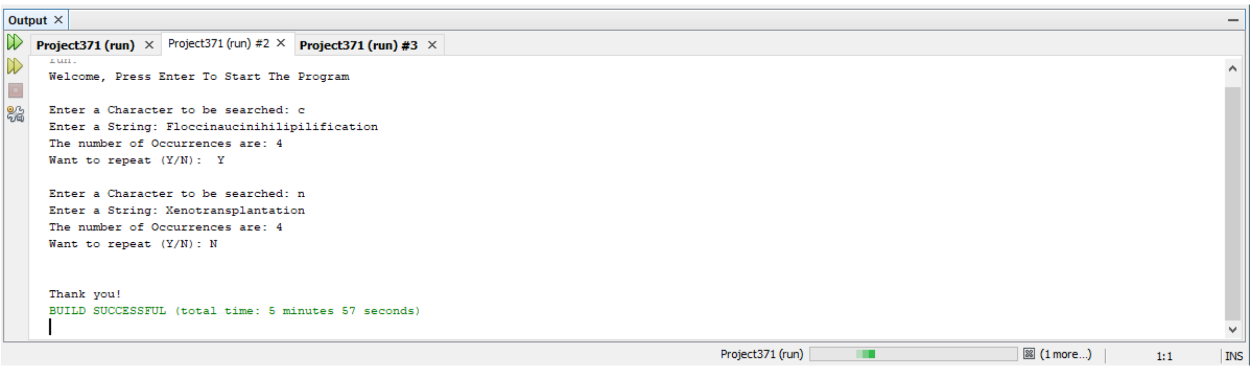
Client



```
run:
Welcome, Press Enter To Start The Program

Enter a Character to be searched: c
Enter a String: Floccinaucinihilipilification
The number of Occurrences are: 4
Want to repeat (Y/N):  Y

Enter a Character to be searched:
```

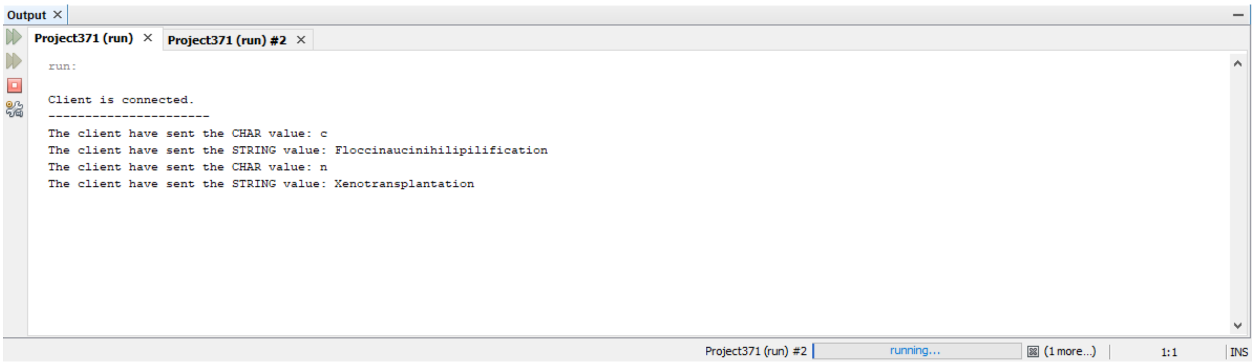
6- Repetition to prove persistent

## Client



```
run:
Welcome, Press Enter To Start The Program

Enter a Character to be searched: c
Enter a String: Floccinaucinihilipilification
The number of Occurrences are: 4
Want to repeat (Y/N):  Y

Enter a Character to be searched: n
Enter a String: Xenotransplantation
The number of Occurrences are: 4
Want to repeat (Y/N): N


Thank you!
BUILD SUCCESSFUL (total time: 5 minutes 57 seconds)
```

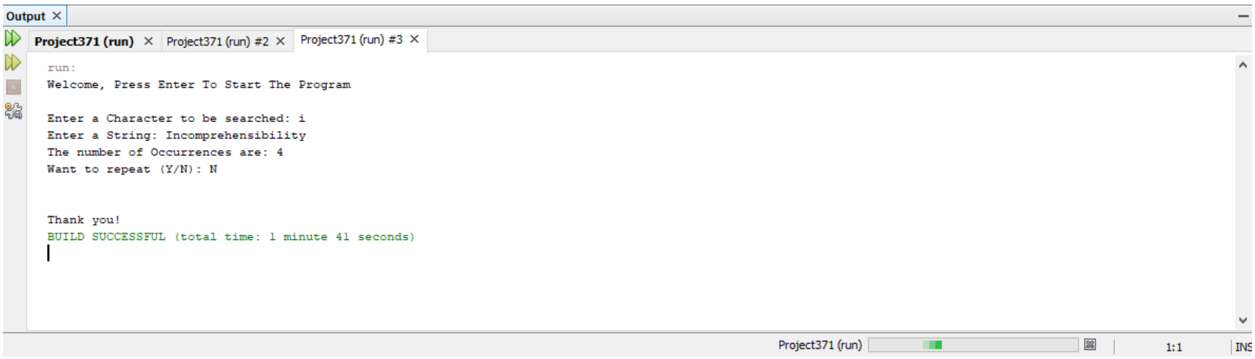
## Server



```
run:

Client is connected.
----------------------
The client have sent the CHAR value: c
The client have sent the STRING value: Floccinaucinihilipilification
The client have sent the CHAR value: n
The client have sent the STRING value: Xenotransplantation
```

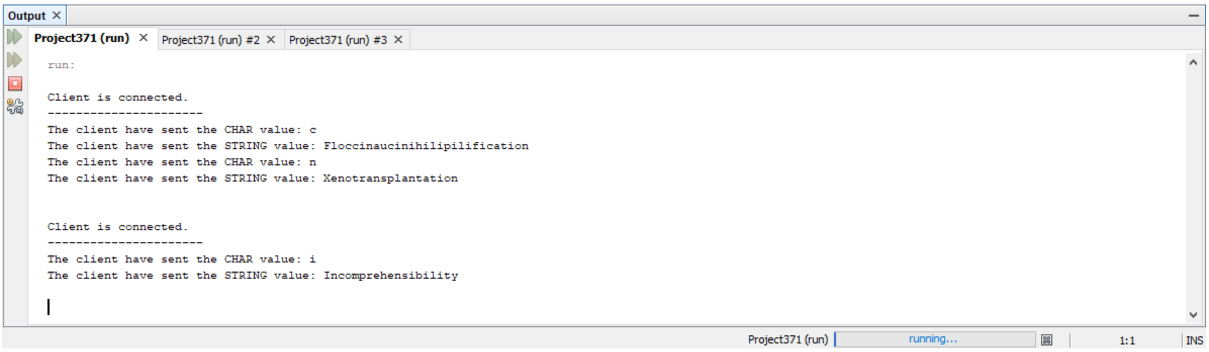
7- Another Run to prove that the 1st client was disconnected

## Client



```
run:
Welcome, Press Enter To Start The Program

Enter a Character to be searched: i
Enter a String: Incomprehensibility
The number of Occurrences are: 4
Want to repeat (Y/N): N


Thank you!
BUILD SUCCESSFUL (total time: 1 minute 41 seconds)
```

## Server



```
run:

Client is connected.
----------------------
The client have sent the CHAR value: c
The client have sent the STRING value: Floccinaucinihilipilification
The client have sent the CHAR value: n
The client have sent the STRING value: Xenotransplantation


Client is connected.
----------------------
The client have sent the CHAR value: i
The client have sent the STRING value: Incomprehensibility
```