



CS330_Computer Networks Project

TCP-based client server application

Students Names:

Students Names:	Id
Neehal Saleh Almouais	
Rahaf Hamoud Alosaimi	
Renad Saud AL hussain	
Dania Hamed Alsaigh	

Section: 373

12/11/2021

1-Setting up the Programming Environment

The programming language we have chosen in this project is **Java Programming Language** because it is easy to understand and easy to learn and it helps us to do a network communication on it. And we have been using Visual Studio Code (VS Code) and it is a text editor version from Microsoft. Opens source for Windows, Mac OS, and Linux drivers and we use it to write and run our Java code.

How we installed the environment?

- We go to their official website and we downloaded it, we followed the same steps:

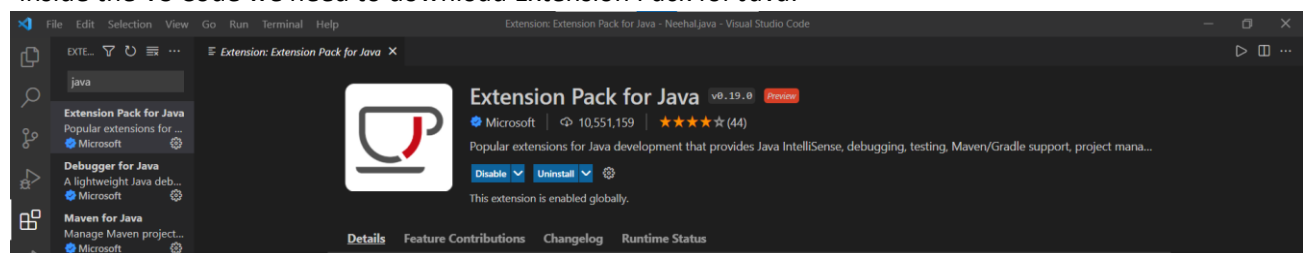
<https://code.visualstudio.com/download>

- Then we need to download JDK "1.8.0_302" to write and run TCP-socket programming and we followed the same steps: <https://www.youtube.com/watch?v=Z4-gVFU5yJU>

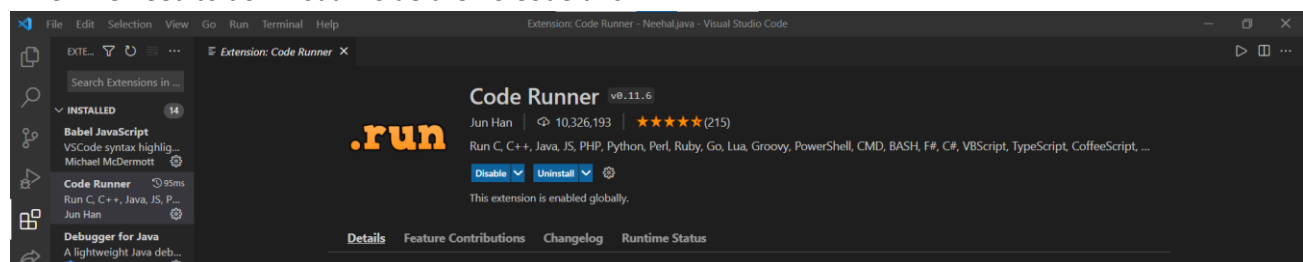
-- And that is our version that we used

```
C:\Users\USER>java -version
openjdk version "1.8.0_302"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_302-b08)
OpenJDK 64-Bit Server VM (Temurin)(build 25.302-b08, mixed mode)
```

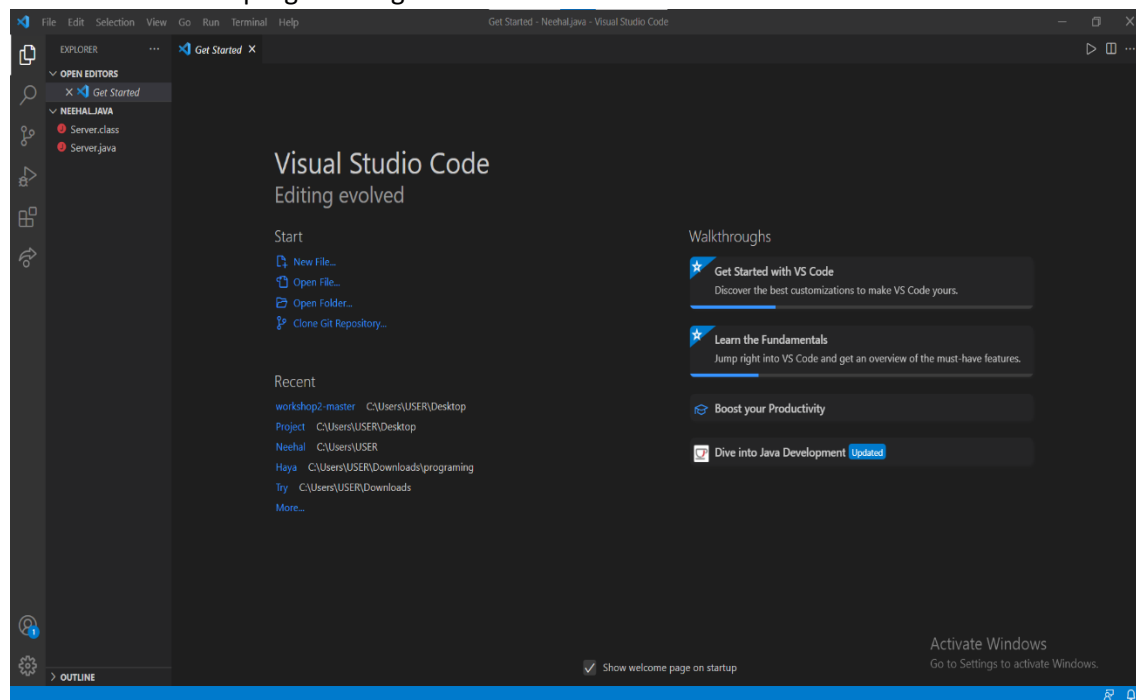
- Inside the VS Code we need to download Extension Pack for Java:



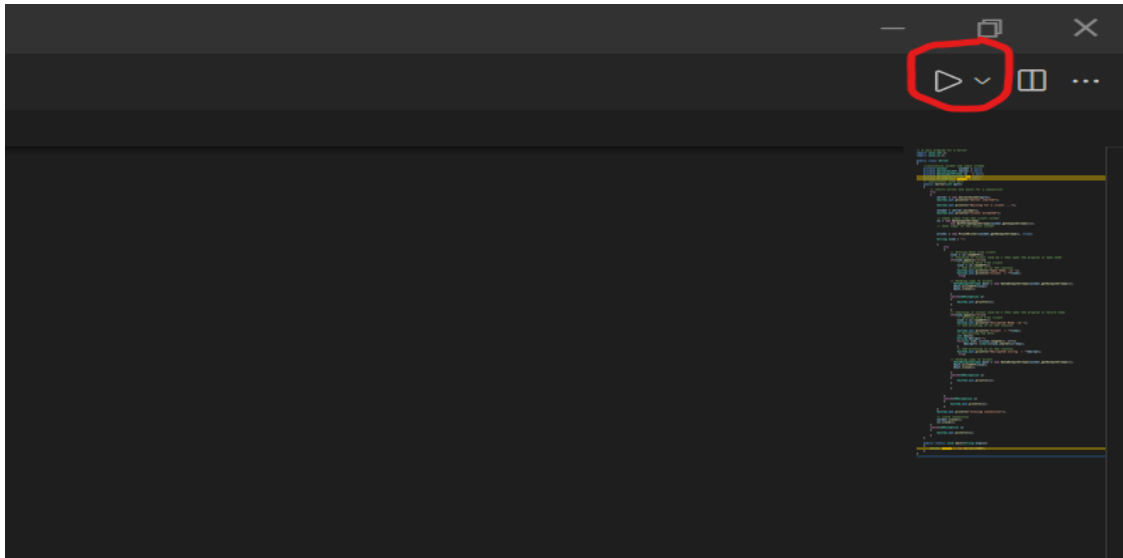
- Then we need to download inside the VS Code this:



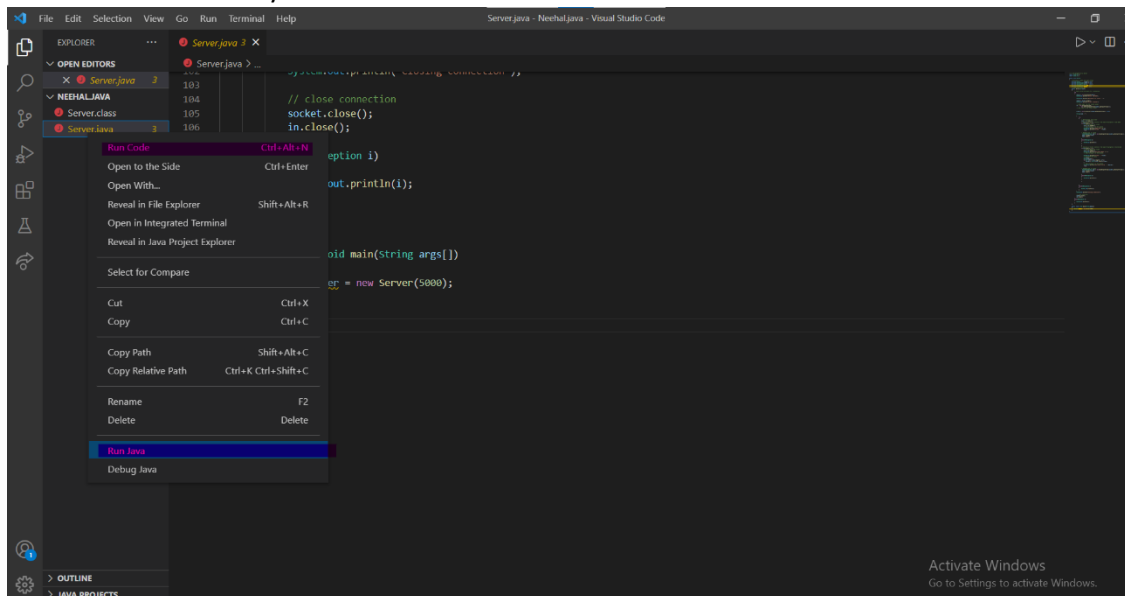
- Screenshot of the programming environment:



how we compile and run our program: we have to option to run



-click left on the class you want to run and chose Run Code OR Run Java



2-Steps for TCP socket programming for client-server connection

We use Java socket programming to implement the client-server communication over TCP protocol [<https://www.javatpoint.com/socket-programming> and <https://www.geeksforgeeks.org/socket-programming-in-java/>]

- The server will create the socket using:

```
18     try
19     {
20         server = new ServerSocket(port);
21         System.out.println("Server started");
22
23         System.out.println("Waiting for a client ...");
24
25         socket = server.accept();
26         System.out.println("client accepted");
27
28         // takes input from the client socket
29         in = new DataInputStream(
30             new BufferedInputStream(socket.getInputStream()));
```

In this part of code, the server program starts by creating a new `ServerSocket` object to listen to a specific port. (`server = new ServerSocket(port);`)

While running the server, it will choose a port that is not used. If the server successfully binds to its port, then the `ServerSocket` object is successfully created, and the server will go to the next step accepting a connection from a client (`socket = server.accept();`).

And when client is connected start the communication and we wait for till client send input.

- The client will:

```
12     private DataInputStream in = null;
13     // constructor to put ip address and port
14     public Client(String address, int port) throws IOException {
15         // establish a connection
16         try
17         {
18             socket = new Socket(address, port);
19             System.out.println("connected");
20         }
```

Creating a stream socket and connecting it to the specific port number at the specific IP address (`public Client(String address, int port)`) and then start the communication and send input.

3-Steps for setting up the network

We set up two hosts by use two laptops one for a client and one for a server

We used wireless technology

Step 1: Connect two Computers using the same wireless access point

Step 2: We must get the IP address for server by:

1-go to command Prompt

2-write "ipconfig"

3- Then we will find it here

```
Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::6432:c695:e0f9:22a1%20
IPv4 Address. . . . . : 172.20.10.9
Subnet Mask . . . . . : 255.255.255.240
Default Gateway . . . . . : 172.20.10.1

Ethernet adapter {GUID} Bluetooth 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
```

Step 3: We put the IP in the client code and then run it

Now communication will complete successfully.

4- Codes and comments:

Code of server side:

```
Go Run Terminal Help Server.java - Neehal.java - Visual Studio Code
Server.java 3 X
Server.java > ...
1
2 // A Java program for a Server
3 import java.net.*;
4 import java.io.*;
5
6 public class Server
7 {
8     //Initialize socket and input stream
9     private Socket socket = null;
10    private ServerSocket server = null;
11    private DataInputStream in = null;
12    private DataOutputStream out = null;
13    private PrintWriter writer = null;
14    // constructor with port
15    public Server(int port)
16    {
17        // starts server and waits for a connection
18        try
19        {
20            server = new ServerSocket(port);
21            System.out.println("Server started");
22
23            System.out.println("Waiting for a client ...");
24
25            socket = server.accept();
26            System.out.println("client accepted");
27
28            // takes input from the client socket
29            in = new DataInputStream(
30                new BufferedInputStream(socket.getInputStream()));
31            // Send input to the client socket
32
33
34            writer = new PrintWriter(socket.getOutputStream(), true);
35
36            String line = "";
37
38            {
39                try
40                {
41                    // Getting Data from client
42                    line = in.readUTF();
43                    // Checking if Client send me 1 then open the program in open mode
44                    if(line.equals("1")){
45                        // Getting Data from client
46                        line = in.readUTF();
47                        // And printing it on the console.
48                        System.out.println("Open Mode :\n ");
49                        System.out.println("client = "+line);
50                        try{
51
52                            // Sending copy to client
53                            DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
54                            dout.writeUTF(line);
55                            dout.close();
56
57                        }
58                        catch(IOException i)
59                        {
60                            System.out.println(i);
61                        }
62
63                    }
64                    // Checking if Client send me 2 then open the program in Secure mode
65                    if(line.equals("2")){
66                        // Getting data from client
67                        line = in.readUTF();
68                        System.out.println("Encrypted Mode :\n ");
69                        // And printing it on the console.
70
71                        System.out.println("Client = "+line);
72                        // Decrypting the data
73                        int key=2;
74                        String decrypts="";
75                    }
76                }
77            }
78        }
79    }
80}
```

```
Go Run Terminal Help Server.java - Neehal.java - Visual Studio Code
Server.java 3 X
Server.java > ...
73         int key=2;
74         String decrypt="";
75         for(int i=0; i<line.length(); i++){
76             decrypt+= (char)(line.charAt(i)-key);
77         }
78         // And printing it on the console.
79         System.out.println("Decrypted string = "+decrypt);
80         try{
81
82             // Sending copy to Client
83             DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
84             dout.writeUTF(line);
85             dout.close();
86
87         }
88         catch(IOException i)
89         {
90             System.out.println(i);
91         }
92
93     }
94
95
96
97     catch(IOException i)
98     {
99         System.out.println(i);
100     }
101 }
102 System.out.println("Closing connection");
103
104 // close connection
105 socket.close();
106 in.close();
107 }
108 catch(IOException i)
109 {
110     System.out.println(i);
111 }
```

Ln 5, Col 1 Spaces

```
Go Run Terminal Help Server.java - Neehal.java - Visual Studio Code
Server.java 3 X
Server.java > ...
86
87     }
88     catch(IOException i)
89     {
90         System.out.println(i);
91     }
92
93 }
94
95
96
97 catch(IOException i)
98 {
99     System.out.println(i);
100 }
101 }
102 System.out.println("Closing connection");
103
104 // close connection
105 socket.close();
106 in.close();
107 }
108 catch(IOException i)
109 {
110     System.out.println(i);
111 }
112 }
113
114 Run | Debug
115 public static void main(String args[])
116 {
117     Server server = new Server(5000);
118 }
119 }
```

Ln 5, Col 1 Spaces

Code for client side:

```
Go Run Terminal Help Client.java - Neehal.java - Visual Studio Code
Client.java 6 X
Client.java > Client > Client(String, int)
1 // A Java program for a Client
2 import java.net.*;
3 import java.io.*;
4 import java.util.Scanner;
5
6 public class Client
7 {
8     // initialize socket and input output streams
9     private Socket socket = null;
10    private DataInputStream input = null;
11    private DataOutputStream out = null;
12    private DataInputStream in = null;
13    // constructor to put ip address and port
14    public Client(String address, int port) throws IOException {
15        // establish a connection
16        try
17        {
18            socket = new Socket(address, port);
19            System.out.println("Connected");
20
21            // takes input from terminal
22            input = new DataInputStream(System.in);
23
24            // sends output to the socket
25            out = new DataOutputStream(socket.getOutputStream());
26            // sends Input to the socket
27            in = new DataInputStream(socket.getInputStream());
28        }
29        catch(UnknownHostException u)
30        {
31            System.out.println(u);
32        }
33        catch(IOException i)
34        {
35            System.out.println(i);
36        }
37    }
38 }
```

```
Go Run Terminal Help Client.java - Neehal.java - Visual Studio Code
Client.java 6 X
Client.java > Client > Client(String, int)
38
39 // string to read message from input
40 String line = "";
41 // Menu printing and taking the input
42 System.out.println("Press 1 for Open Mode");
43 System.out.println("Press 2 for secure Mode");
44 System.out.println("Press 3 for Quit Application");
45 int choice=0;
46 // Taking input from user
47 Scanner myObj = new Scanner(System.in);
48 int userInput = myObj.nextInt();
49
50 // Sending input to client it means server have to open program in open mode or secure mode
51 out.writeUTF(String.valueOf(userinput));
52
53 // Checking if the input is 1 then open the program in open mode
54 if(userinput==1){
55     // Error handling
56     try
57     {
58         System.out.println("Enter Input");
59         // Taking input from user and sending to Server
60         line = input.readLine();
61         out.writeUTF(line);
62     }
63     catch(IOException i)
64     {
65         System.out.println(i);
66     }
67     try {
68         // Getting copy from Server
69
70         DataInputStream dis = new DataInputStream(socket.getInputStream());
71         String copydata = (String) dis.readUTF();
72         System.out.println("Copy Got from Server : " + copydata);
73
74         dis.close();
75     }
76 }
```



```
Go Run Terminal Help Client.java - Neehal.java - Visual Studio Code
Client.java 6 X
Client.java > Client > Client(String, int)
73
74     dis.close();
75
76 }
77
78 catch(IOException i)
79 {
80     System.out.println(i);
81 }
82
83 }
84 // Checking if the input is 2 then open the program in secure mode
85
86 else if(userinput==2){
87     // Encryption Key
88     int key = 2;
89     // Taking input from user
90     System.out.println("Enter Input");
91     line = input.readLine();
92     // Encrypting the string using key
93     String encrypt="";
94     for(int i =0; i<line.length(); i++){
95         encrypt+= (char) (line.charAt(i)+key);
96     }
97     // Error Handling
98
99     try
100     {
101         // Sending input to server and server will decode their.
102         out.writeUTF(encrypt);
103     }
104     catch(IOException i)
105     {
106         System.out.println(i);
107     }
108 }
109 try {
110     // Getting copy from Server
```

```
Go Run Terminal Help Client.java - Neehal.java - Visual Studio Code
Client.java 6 X
Client.java > Client > Client(String, int)
110 // Getting copy from Server
111
112     DataInputStream dis = new DataInputStream(socket.getInputStream());
113     String copydata = (String) dis.readUTF();
114     System.out.println("Copy Got from Server : " + copydata);
115
116     dis.close();
117
118 }
119
120 catch(IOException i)
121 {
122     System.out.println(i);
123 }
124
125 }
126
127 // close the connection
128 try
129 {
130     input.close();
131     out.close();
132     socket.close();
133 }
134 catch(IOException i)
135 {
136     System.out.println(i);
137 }
138 }
139 }
140
141 Run | Debug
142 public static void main(String args[]) throws IOException {
143     Client client = new Client("127.0.0.1", 5000);
144 }
145 }
```

5- Snapshots of the application outputs:

1. Open mode

```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL
[Running] cd "c:\Users\USER\Desktop\Project\Neehal\Neehal.java\" && javac Server.java && java Server
Server started
Waiting for a client ...
Client accepted
Open Mode :

Client = this is our try for open mode
Closing connection

PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Press 1 for Open Mode
Press 2 for secure Mode
Press 3 for Quit Application
1
Enter Input
this is our try for open mode
Copy Got from Server : this is our try for open mode
PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java>
```

2. Secure mode

```
[Running] cd "c:\Users\USER\Desktop\Project\Neehal\Neehal.java\" && javac Server.java && java Server
Server started
Waiting for a client ...
Client accepted
Encrypted Mode :

Client = vjku"ku"qwt"vt{"vq"ugewtg"oqfg
Decrypted string = this is our try to secure mode
Closing connection

[Done] exited with code=0 in 47.585 seconds

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL powershell +
PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Press 1 for Open Mode
Press 2 for secure Mode
Press 3 for Quit Application
2
Enter Input
this is our try to secure mode
Copy Got from Server : vjku"ku"qwt"vt{"vq"ugewtg"oqfg
PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java>
```

3. Quit Mode

```
PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL
[Running] cd "c:\Users\USER\Desktop\Project\Neehal\Neehal.java\" && javac Server.java && java Server
Server started
Waiting for a client ...
Client accepted
Closing connection

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java> java Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected
Press 1 for Open Mode
Press 2 for secure Mode
Press 3 for Quit Application
3
PS C:\Users\hkb-0\OneDrive\حظي\Neehal.java>
```

6- Problems and solutions:

Problem 1: we faced some basic programming errors like string comparison, like when we was compare the string with == operator but it was not working.

Solution: we Got the solution from stack overflow.

Problem 2: at the first we do not know how we know the IP address of our laptops

Solution: we watch this video (<https://www.youtube.com/watch?v=pbfE4ypNxLI>)

Problem 3: We had difficulty encrypting the message between the client and server, we tried several methods that didn't work

Solution: We read this article and benefited a lot, and I found the most useful method

References:

1. https://www.w3schools.com/java/java_user_input.asp
2. <https://www.javatpoint.com/socket-programming>
3. <https://www.pearsonitcertification.com/articles/article.aspx?p=1680706>