كلية علوم الحاسب والمعلومات
College of Computer and Information Sciences

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

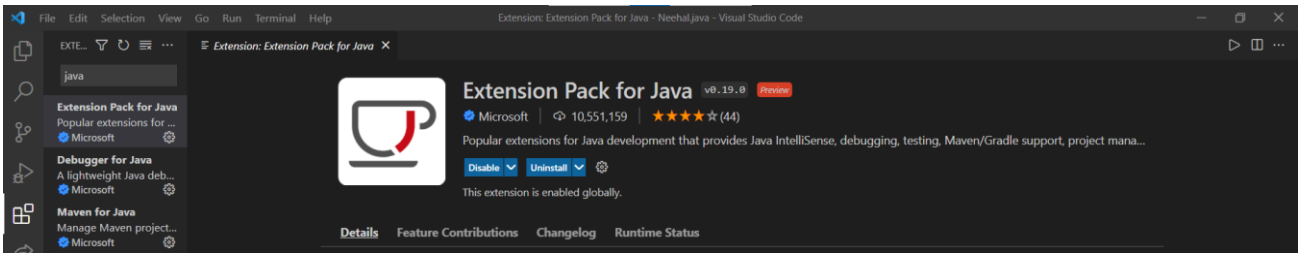
1-Setting up the Programming Environment

The programing language we have chosen in this project is **Java Programming Language** because it is easy to understand and easy to learn and it is help us to do a network communication on it.
And we have been used 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.

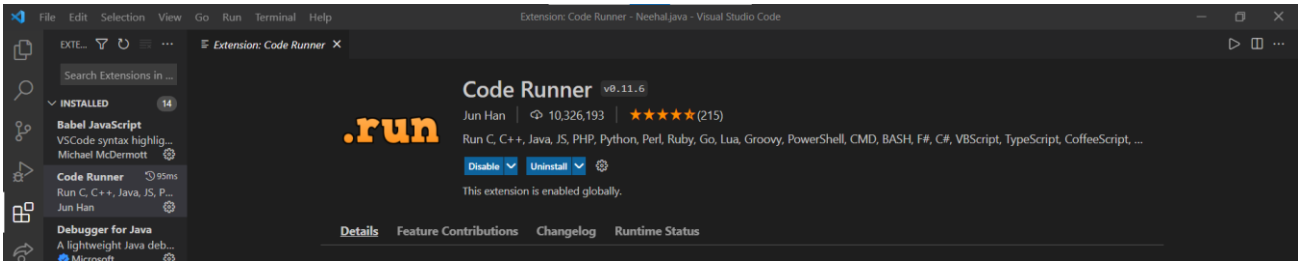
how we installed the environment?
- we go to their official website and we downloaded it, we followed the same steps:
https://code.visualstudio.com/download
-Then we need to download JDK"1.8.0_302" to write and run TCP-Socket Programming and we followed the same steps: https://www.youtube.com/watch?v=Z4-gVFU5yJU
--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)
```

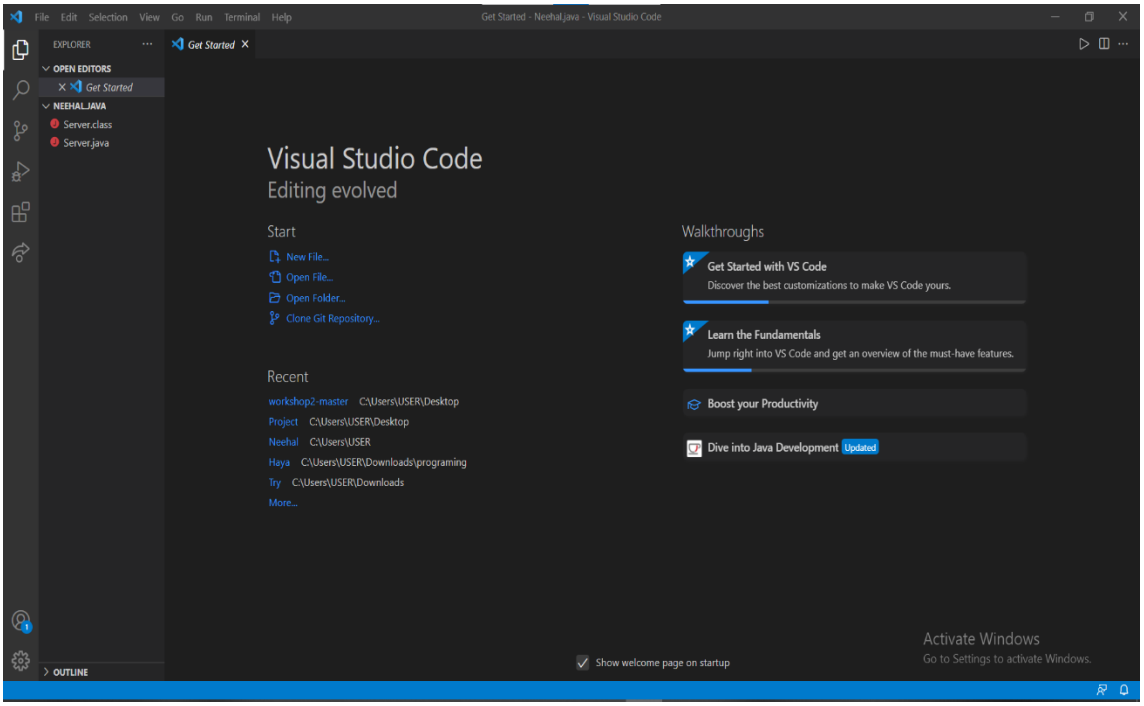
-inside the VS Code we need to download Extension Pack for Java:



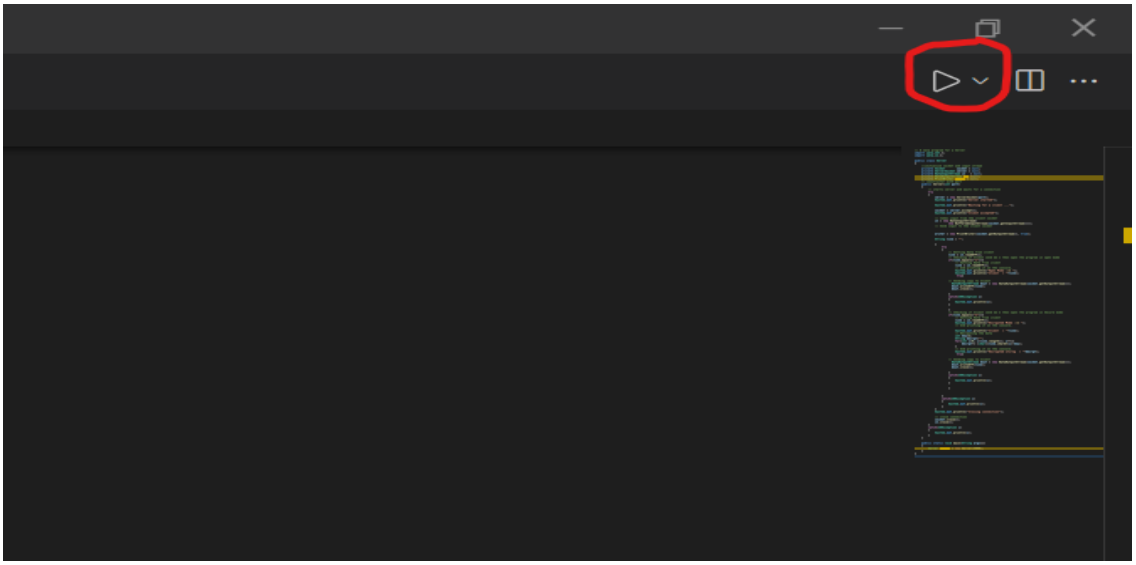
-Then we need to download inside the VS Code this:



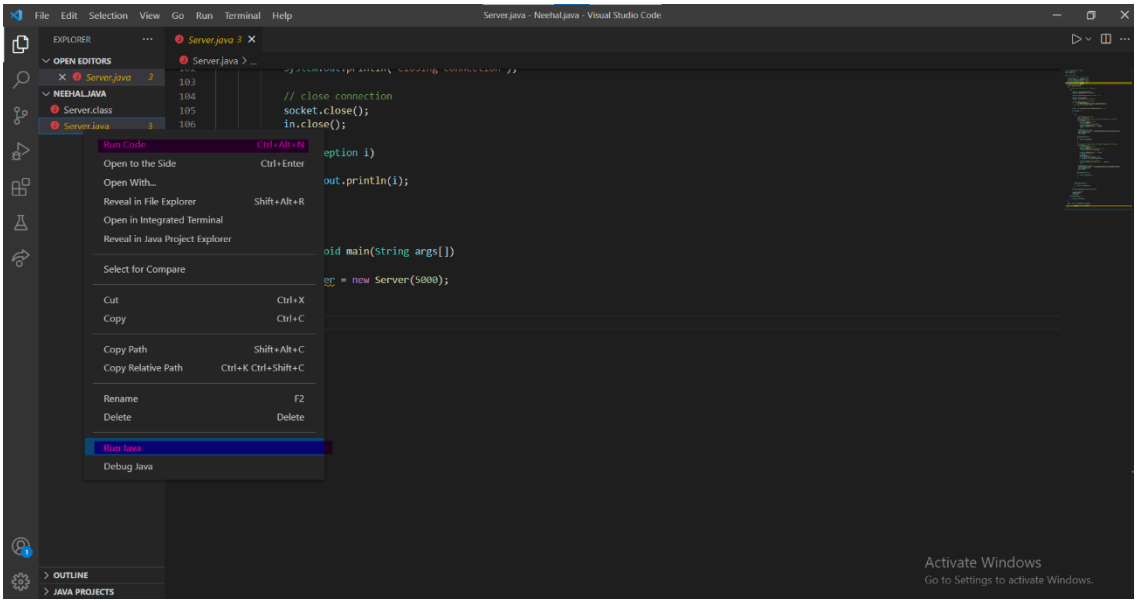
- Screenshot of the programming environment:



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:

```
try
{
    server = new ServerSocket(port);
    System.out.println("Server started");

    System.out.println("Waiting for a client ...");

    socket = server.accept();
    System.out.println("Client accepted");

    // takes input from the client socket
    in = new DataInputStream(
```

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 ……:

```
// constructor to put ip address and port
public Client(String address, int port) throws IOException {
    // establish a connection
    try
    {
        socket = new Socket(address, port);
        System.out.println("Connected");
```

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 tow laptops one for a client and one for a server

We used wireless technology

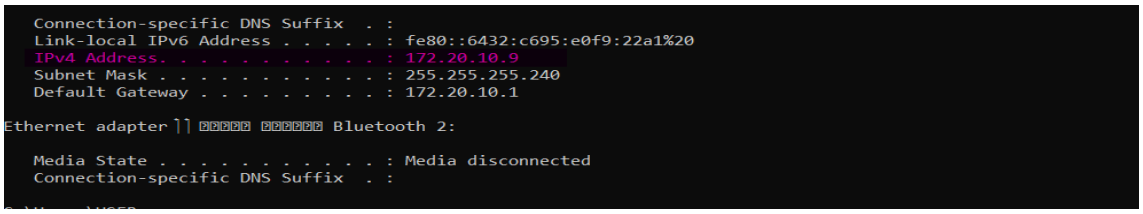
Step 1: Connect two Computers using the same wireless access point

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

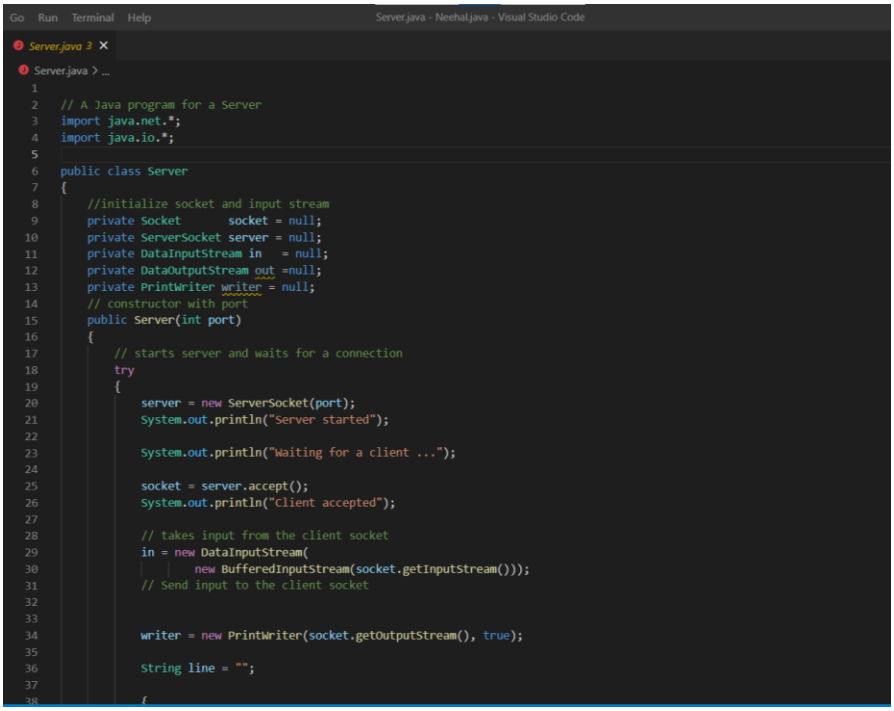


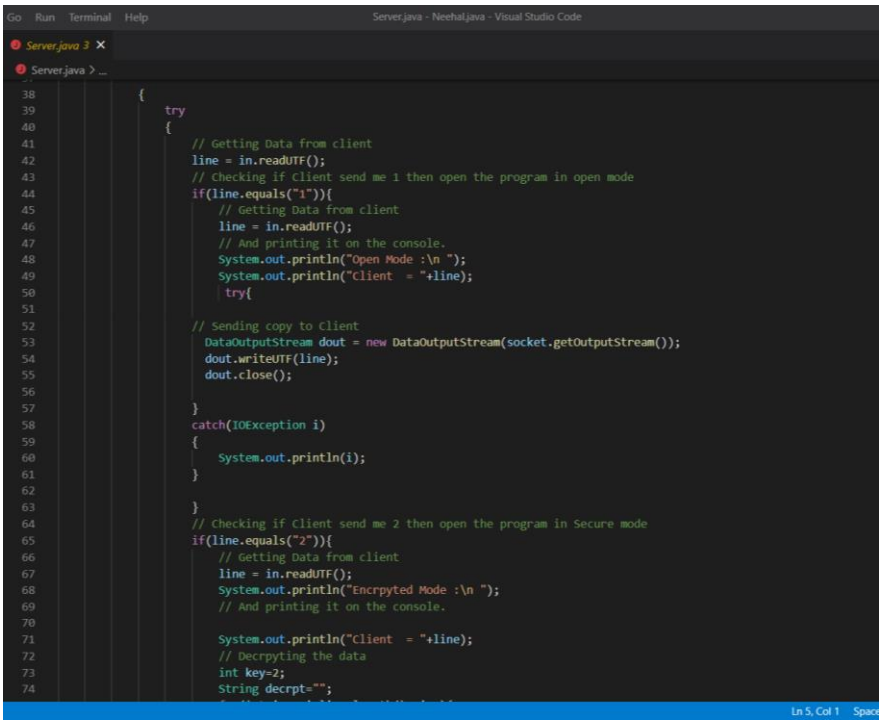
Step 3: We put the IP in the client code and then run it

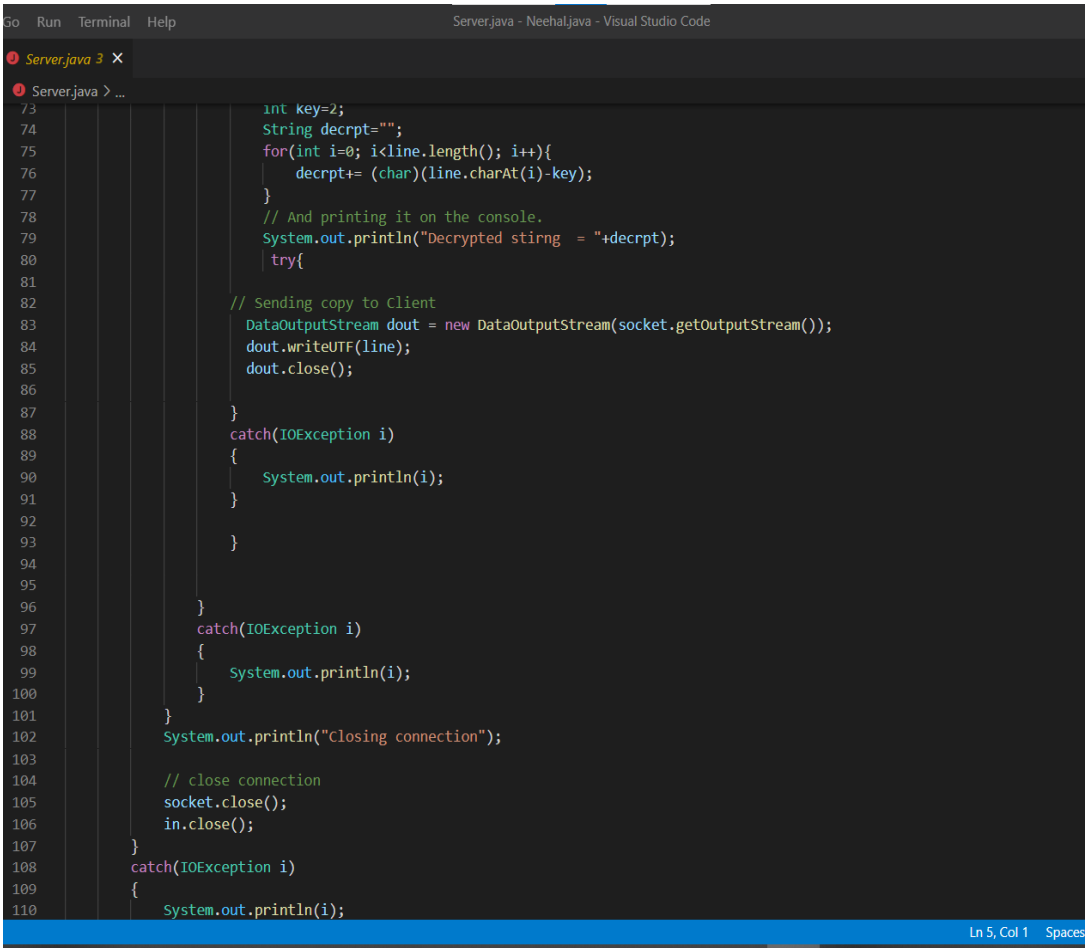
Now communication will complete successfully.

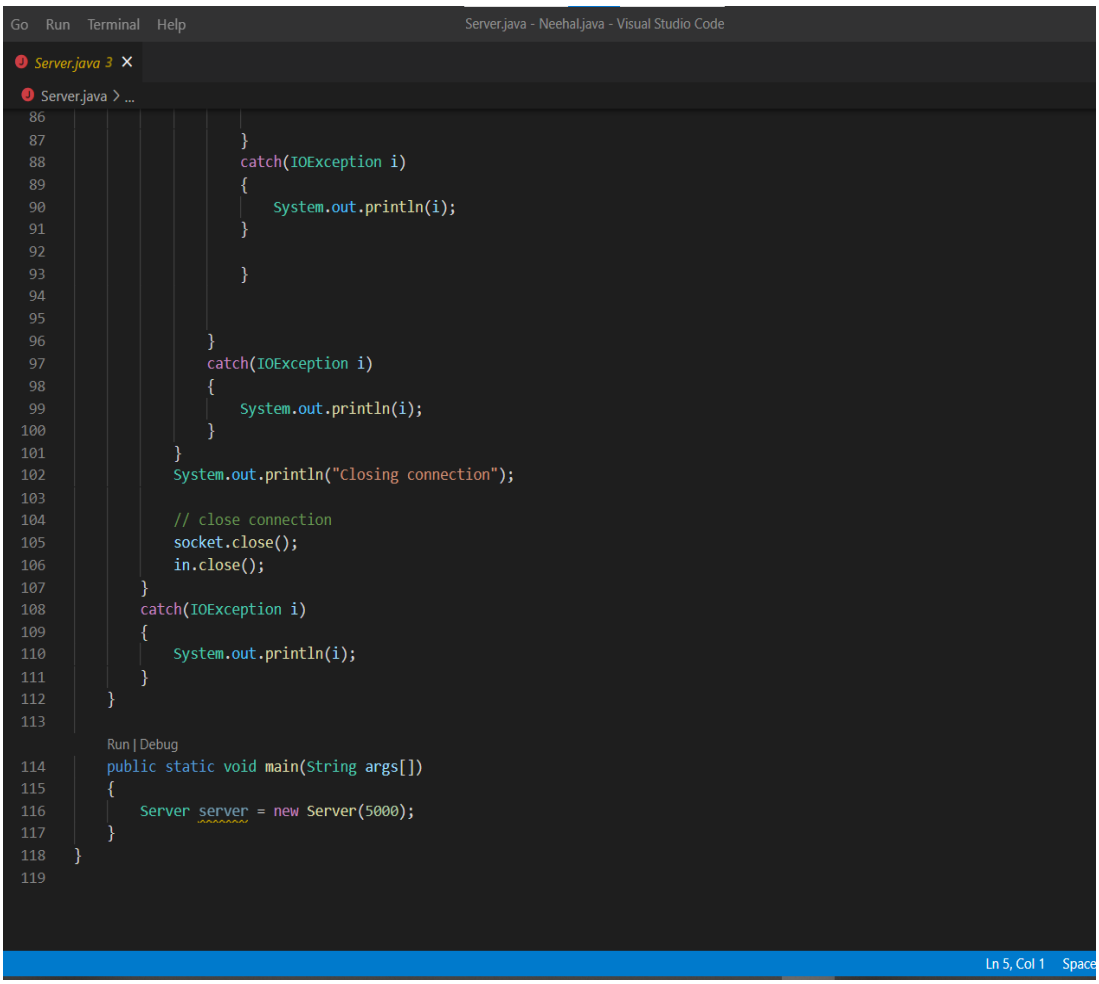
4- Codes and comments:

Code of server side:






Go Run Terminal Help
Server.java - Neehal.java - Visual Studio Code
Server.java 3
Server.java > ...
int key=2;
String decrpt="";
for(int i=0; i<line.length(); i++){
decrpt+= (char)(line.charAt(i)-key);
}
// And printing it on the console.
System.out.println("Decrypted stirng = "+decrpt);
try{
// Sending copy to Client
DataOutputStream dout = new DataOutputStream(socket.getOutputStream());
dout.writeUTF(line);
dout.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
catch(IOException i)
{
System.out.println(i);
}
}
System.out.println("Closing connection");
// close connection
socket.close();
in.close();
}
catch(IOException i)
{
System.out.println(i);
Ln 5, Col 1
Spaces


Go Run Terminal Help
Server.java - Neehal.java - Visual Studio Code
Server.java 3
Server.java > ...
}
catch(IOException i)
{
System.out.println(i);
}
}
catch(IOException i)
{
System.out.println(i);
}
}
System.out.println("Closing connection");
// close connection
socket.close();
in.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
Run | Debug
public static void main(String args[])
{
Server server = new Server(5000);
}
}
Ln 5, Col 1
Space

Code for client side:

Go Run Terminal Help — Client.java - Neehal.java - Visual Studio Code

Client.java 6 ×

Client.java > Client > Client(String, int)

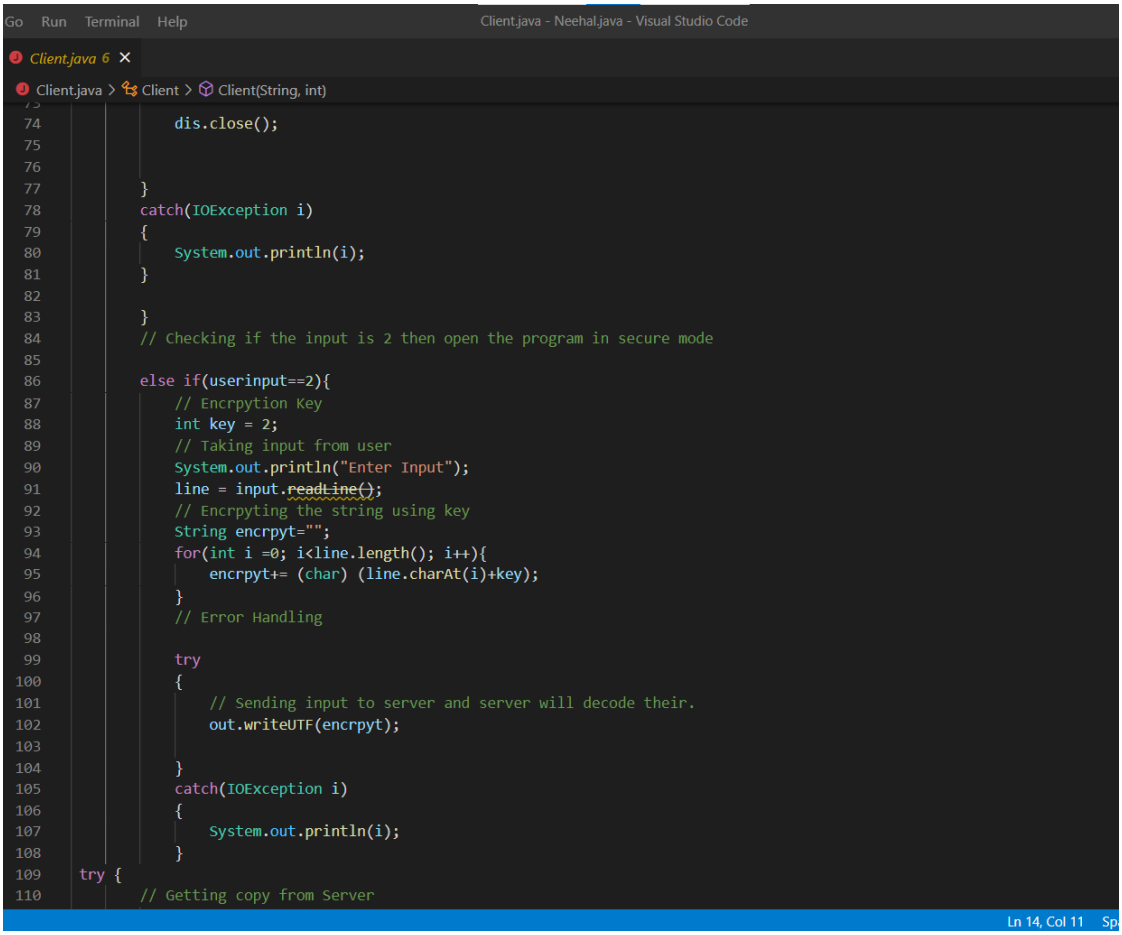
```java
// A Java program for a Client
import java.net.*;
import java.io.*;
import java.util.Scanner;

public class Client
{
    // initialize socket and input output streams
    private Socket socket        = null;
    private DataInputStream input = null;
    private DataOutputStream out     = null;
    private DataInputStream in = null;
    // constructor to put ip address and port
    public Client(String address, int port) throws IOException {
        // establish a connection
        try
        {
            socket = new Socket(address, port);
            System.out.println("Connected");

            // takes input from terminal
            input = new DataInputStream(System.in);

            // sends output to the socket
            out = new DataOutputStream(socket.getOutputStream());
            // sends Input to the socket
            in = new DataInputStream(socket.getInputStream());

        }
        catch(UnknownHostException u)
        {
            System.out.println(u);
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
```

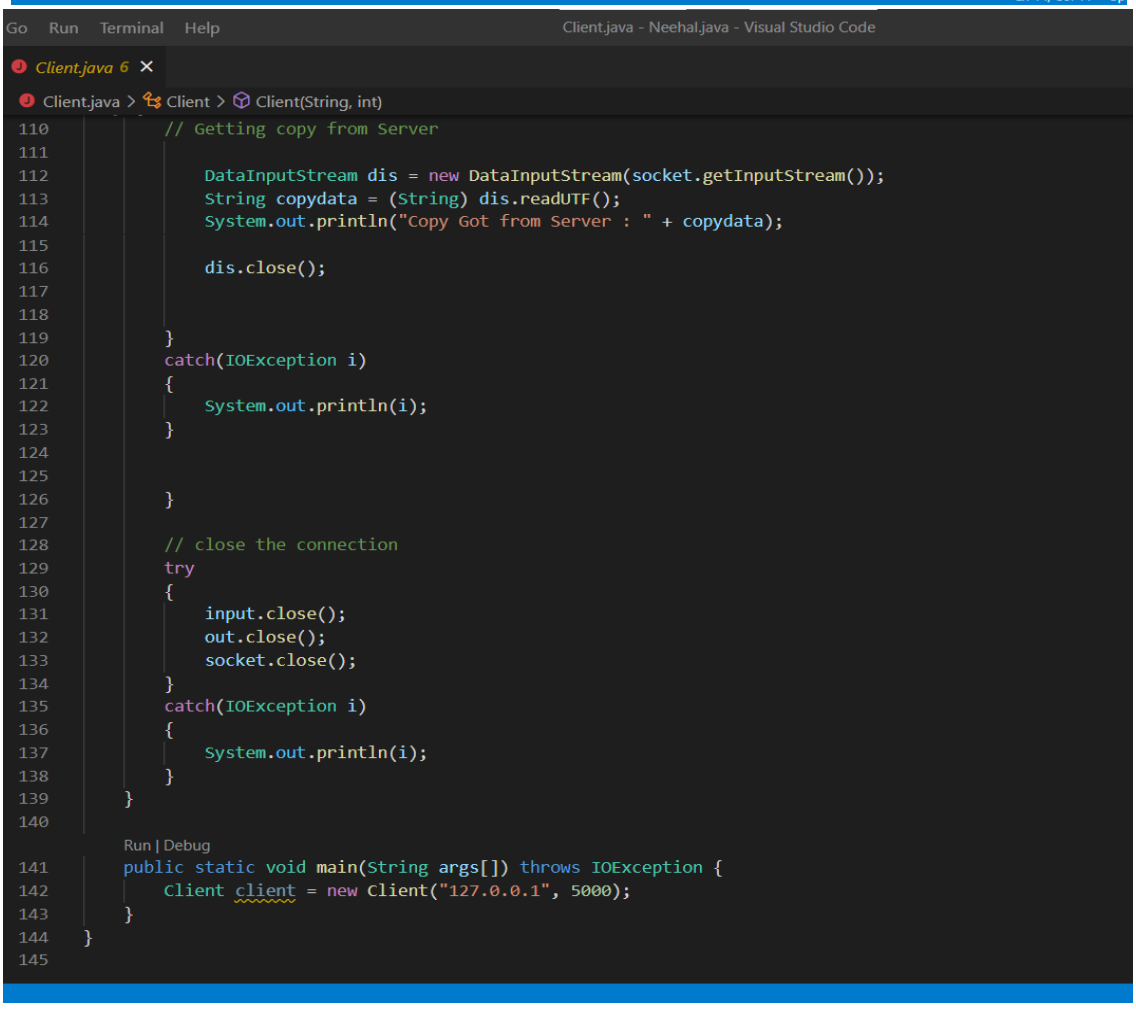
Go Run Terminal Help — Client.java - Neehal.java - Visual Studio Code

Client.java 6 ×

Client.java > Client > Client(String, int)

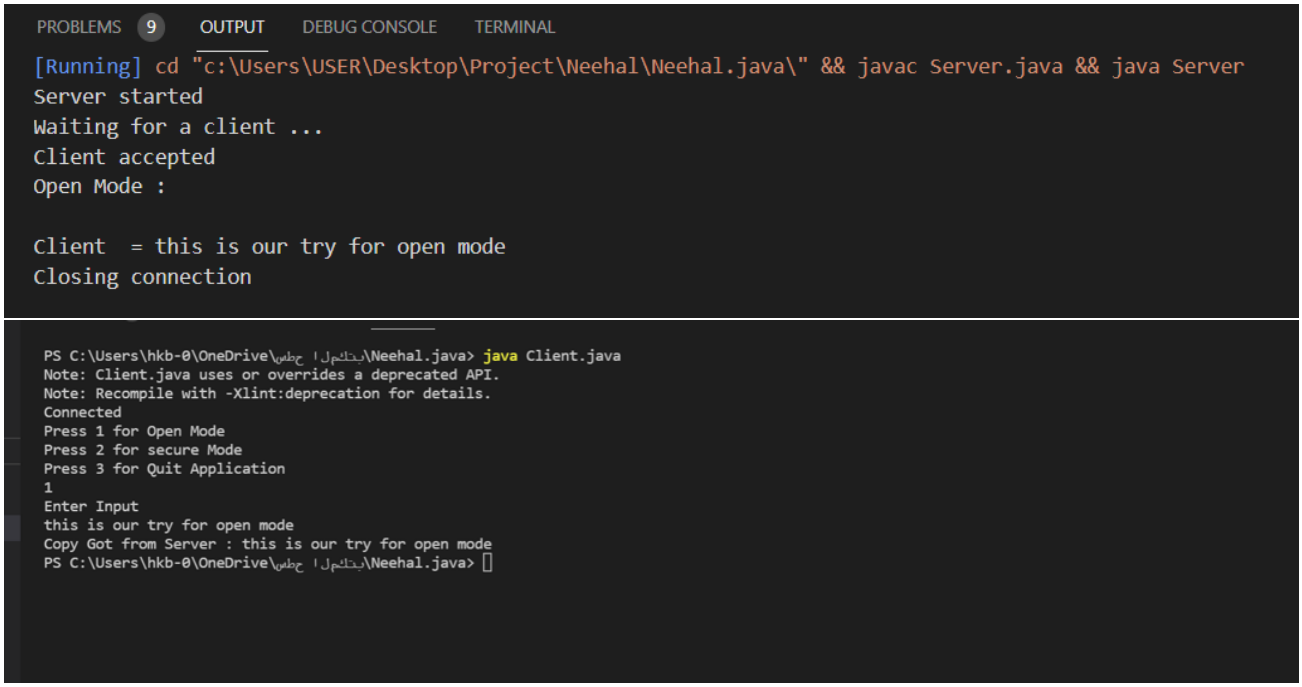
```java
        // string to read message from input
        String line = "";
        // Menu printing and taking the input
        System.out.println("Press 1 for Open Mode");
        System.out.println("Press 2 for secure Mode");
        System.out.println("Press 3 for Quit Application");
        int choice=0;
        // Taking input from user
        Scanner myObj = new Scanner(System.in);
        int userinput = myObj.nextInt();

        // Sending input to client it means server have to open program in open mode or secure mode
        out.writeUTF(String.valueOf(userinput));

        // Checking if the input is 1 then open the program in open mode
        if(userinput==1){
            // Error handling
            try
            {
                System.out.println("Enter Input");
                // Taking input from user and sending to Server
                line = input.readLine();
                out.writeUTF(line);
            }
            catch(IOException i)
            {
                System.out.println(i);
            }
             try {
        // Getting copy from Server

            DataInputStream dis = new DataInputStream(socket.getInputStream());
            String copydata = (String) dis.readUTF();
            System.out.println("Copy Got from Server : " + copydata);

            dis.close();
```


Go Run Terminal Help
Client.java - Neehal.java - Visual Studio Code
Client.java 6
Client.java > Client > Client(String, int)
dis.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
// Checking if the input is 2 then open the program in secure mode
else if(userinput==2){
// Encrpytion Key
int key = 2;
// Taking input from user
System.out.println("Enter Input");
line = input.readline();
// Encrpyting the string using key
String encrpyt="";
for(int i =0; i<line.length(); i++){
encrpyt+= (char) (line.charAt(i)+key);
}
// Error Handling
try
{
// Sending input to server and server will decode their.
out.writeUTF(encrpyt);
}
catch(IOException i)
{
System.out.println(i);
}
try {
// Getting copy from Server
Ln 14, Col 11

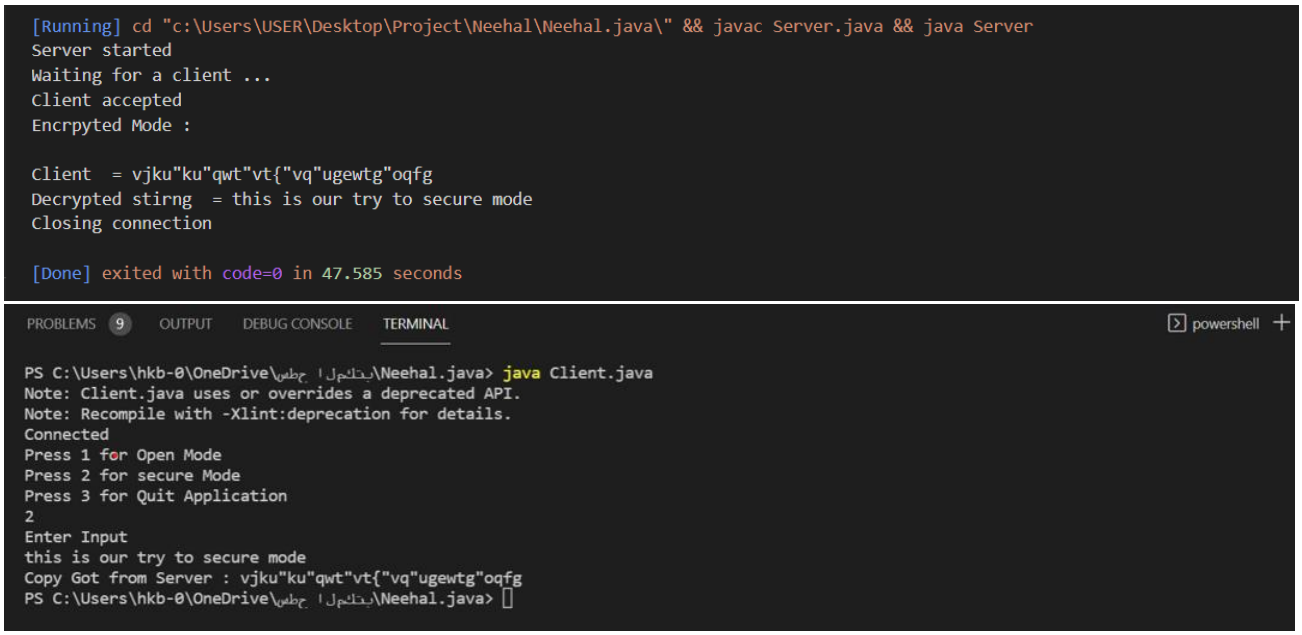

Go Run Terminal Help
Client.java - Neehal.java - Visual Studio Code
Client.java 6
Client.java > Client > Client(String, int)
// Getting copy from Server
DataInputStream dis = new DataInputStream(socket.getInputStream());
String copydata = (String) dis.readUTF();
System.out.println("Copy Got from Server : " + copydata);
dis.close();
}
catch(IOException i)
{
System.out.println(i);
}
// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
Run | Debug
public static void main(String args[]) throws IOException {
Client client = new Client("127.0.0.1", 5000);
}
}

5- Snapshots of the application outputs:

1.Open mode



2. Secure mode



3. Quit Mode

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