Kenana Khalid Turabi

11923807

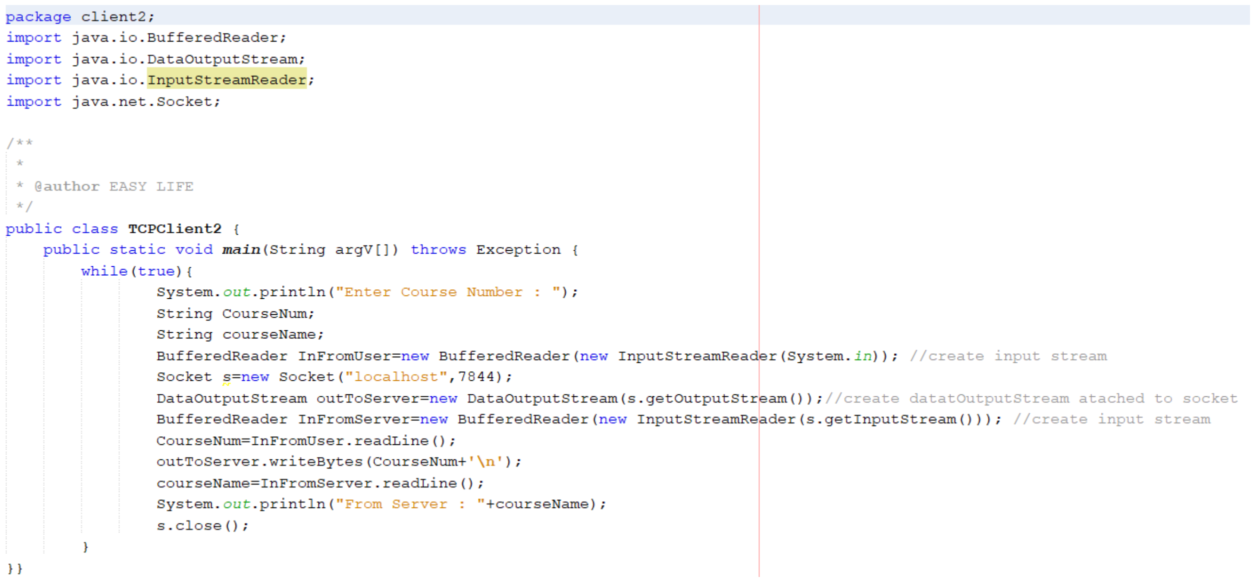
Network Programming HW 1

TCP on the same device (lcalhost)

TCP Notes :

- Each client has a socket , but server has ٢ sockets : server socket (listening socket)-> to establish connection & socket for each client (data communication : sending &receiving packets) after the connection is accepted.
- Each client socket has its own port# (implicitly).but all server sockets has the same port#.
- We have to give the client socket/Tcp socket(src_ip and src_port#).
- Socket works as files we can write on it and read from it .

Client code

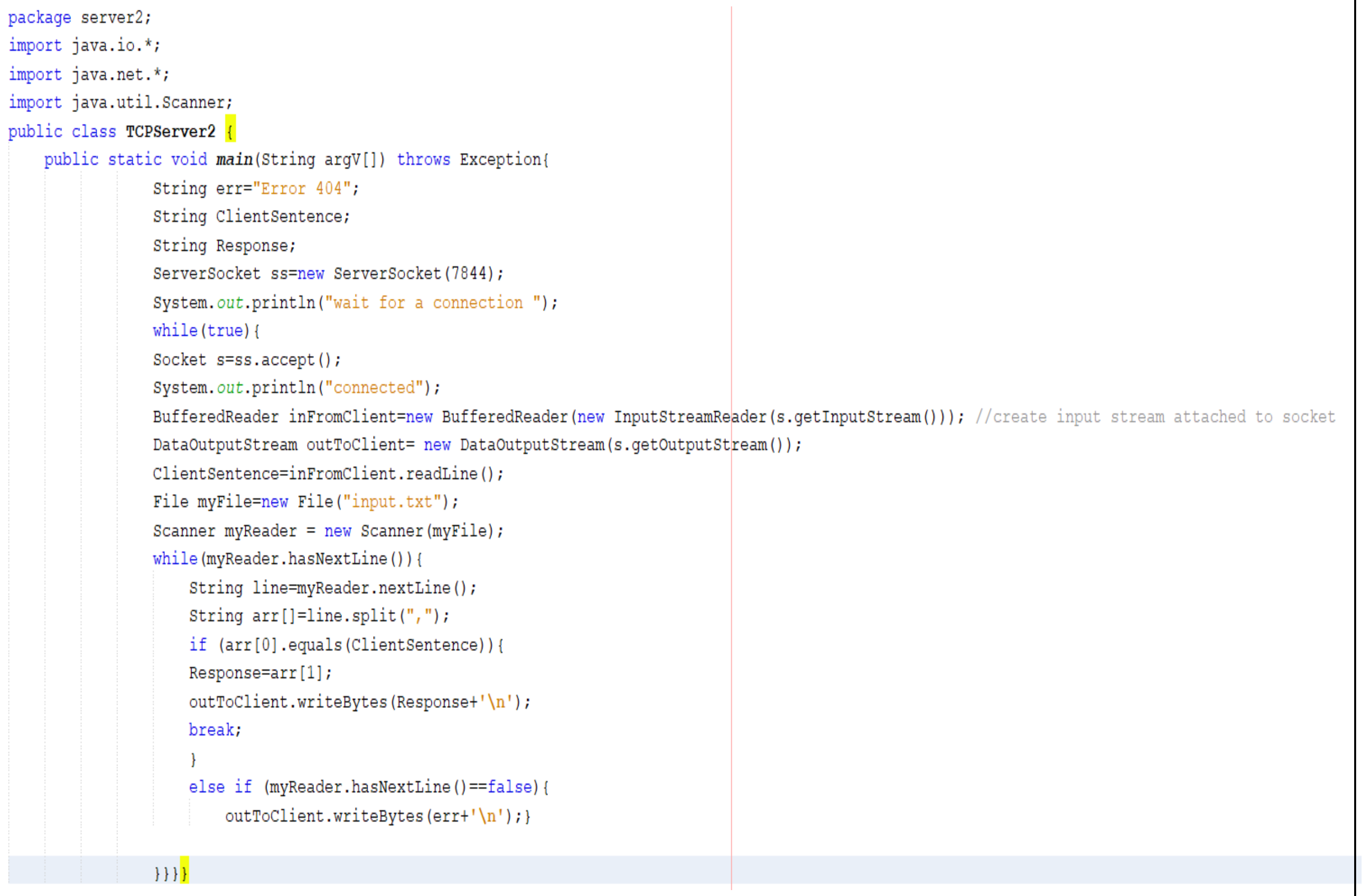


```
package client2;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.Socket;

/**
 *
 * @author EASY LIFE
 */
public class TCPClient2 {
    public static void main(String argV[]) throws Exception {
        while(true){
                System.out.println("Enter Course Number : ");
                String CourseNum;
                String courseName;
                BufferedReader InFromUser=new BufferedReader(new InputStreamReader(System.in)); //create input stream
                Socket s=new Socket("localhost",7844);
                DataOutputStream outToServer=new DataOutputStream(s.getOutputStream());//create datatOutputStream atached to socket
                BufferedReader InFromServer=new BufferedReader(new InputStreamReader(s.getInputStream())); //create input stream
                CourseNum=InFromUser.readLine();
                outToServer.writeBytes(CourseNum+'\n');
                courseName=InFromServer.readLine();
                System.out.println("From Server : "+courseName);
                s.close();
        }
}}
```

TCP_CLIENT_CODE_DESCRIPTION :

1. Create Input stream to read from keyboard (course id)
2. Create Socket and Give it (src_address,port#)
3. Input stream and output stream from socket
4. Read line from Input stream and store it in string
5. Send it to the server(write string on the socket)
6. Read response from server and store it in string
7. Print response on the monitor

- **Server code**

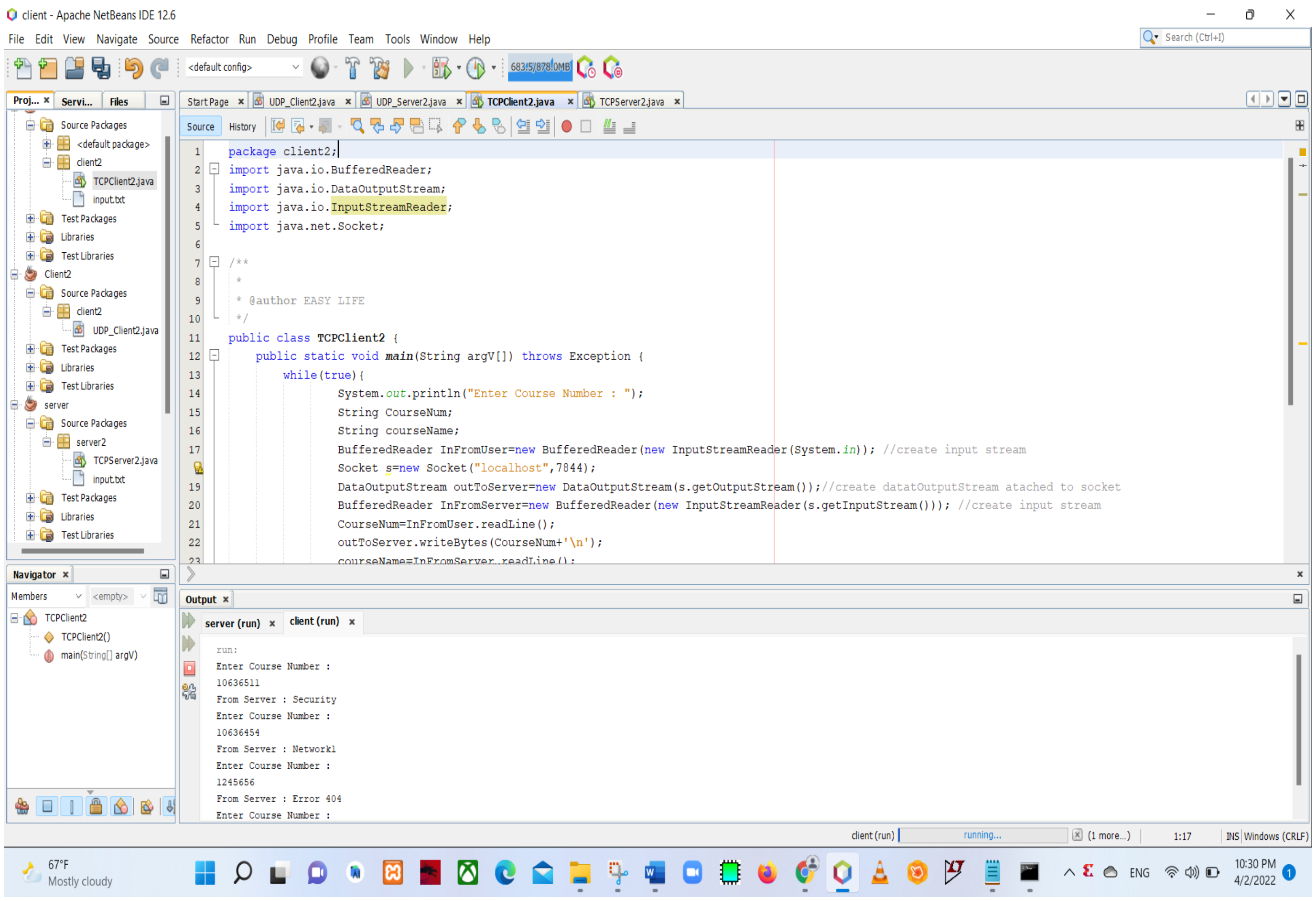


```
package server2;
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class TCPServer2 {
    public static void main(String argV[]) throws Exception{
                String err="Error 404";
                String ClientSentence;
                String Response;
                ServerSocket ss=new ServerSocket(7844);
                System.out.println("wait for a connection ");
                while(true){
                Socket s=ss.accept();
                System.out.println("connected");
                BufferedReader inFromClient=new BufferedReader(new InputStreamReader(s.getInputStream())); //create input stream attached to socket
                DataOutputStream outToClient= new DataOutputStream(s.getOutputStream());
                ClientSentence=inFromClient.readLine();
                File myFile=new File("input.txt");
                Scanner myReader = new Scanner(myFile);
                while(myReader.hasNextLine()){
                    String line=myReader.nextLine();
                    String arr[]=line.split(",");
                    if (arr[0].equals(ClientSentence)){
                    Response=arr[1];
                    outToClient.writeBytes(Response+'\n');
                    break;
                    }
                    else if (myReader.hasNextLine()==false){
                        outToClient.writeBytes(err+'\n');}

                }}}}
```

- **TCP_SERVER_CODE_DESCRIPTION :**

1. Create listening socket (server socket)
2. Create data socket waiting on listening socket to be connected
3. Read from data_socket and encapsulate it in buffered reader
4. Open file
5. Search on the file according to the course number
6. But the course name from file with \n In a response message
7. Write it on the data-socket else write error message

➤ **TCP_Output :**



```
package client2;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.Socket;

/**
 *
 * @author EASY LIFE
 */
public class TCPClient2 {
    public static void main(String argV[]) throws Exception {
        while(true){
                System.out.println("Enter Course Number : ");
                String CourseNum;
                String courseName;
                BufferedReader InFromUser=new BufferedReader(new InputStreamReader(System.in)); //create input stream
                Socket s=new Socket("localhost",7844);
                DataOutputStream outToServer=new DataOutputStream(s.getOutputStream());//create datatOutputStream atached to socket
                BufferedReader InFromServer=new BufferedReader(new InputStreamReader(s.getInputStream())); //create input stream
                CourseNum=InFromUser.readLine();
                outToServer.writeBytes(CourseNum+'\n');
                courseName=InFromServer.readLine();
```

```
run:
Enter Course Number :
10636511
From Server : Security
Enter Course Number :
10636454
From Server : Network1
Enter Course Number :
1245656
From Server : Error 404
Enter Course Number :
```

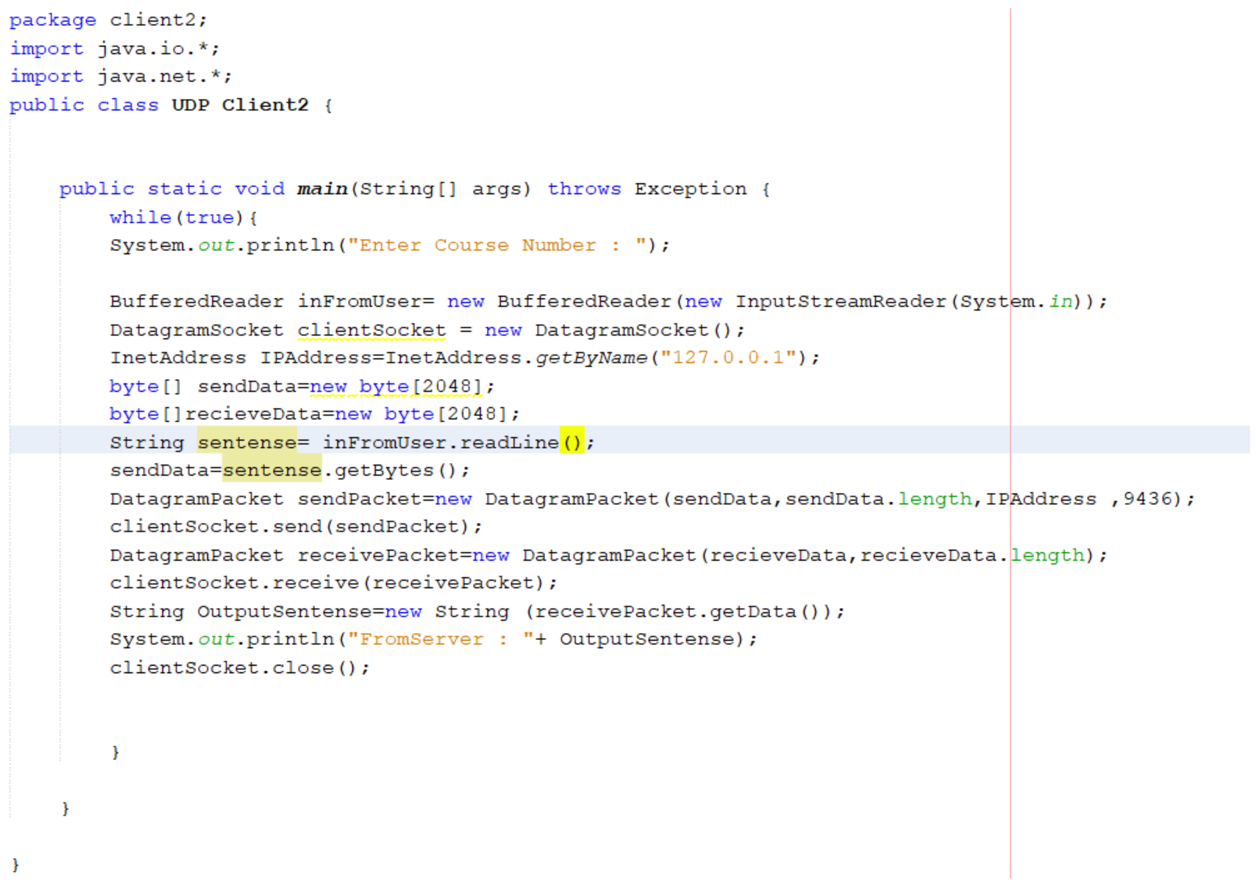
UDP on the same device

➤ **UDP Notes** :

- Client creates socket it type is a Datagram which has data and header . Header contains : des_ip & des_port# explicitly and sec_ip & src_port# implicitly . Server has to receive the datagram and extract sec_ip and src_port# from header to know to whom the server has to response .
- Both server and client must create a sockets , which like interface and works as a file (we can write on it & read from it)
- Server socket we must give it a port# .But client socket the os will give it an available port#.

- Server gets the datagram from the client and get data from it (course id) prepare the response (course name) after searching on the courses file and then create new datagram which contains : src_ip & src_port # (extracted from received datagram) as destination and response data.
- Client read the response and print it on the monitor and then close the socket .

- **UDP Client Code :**



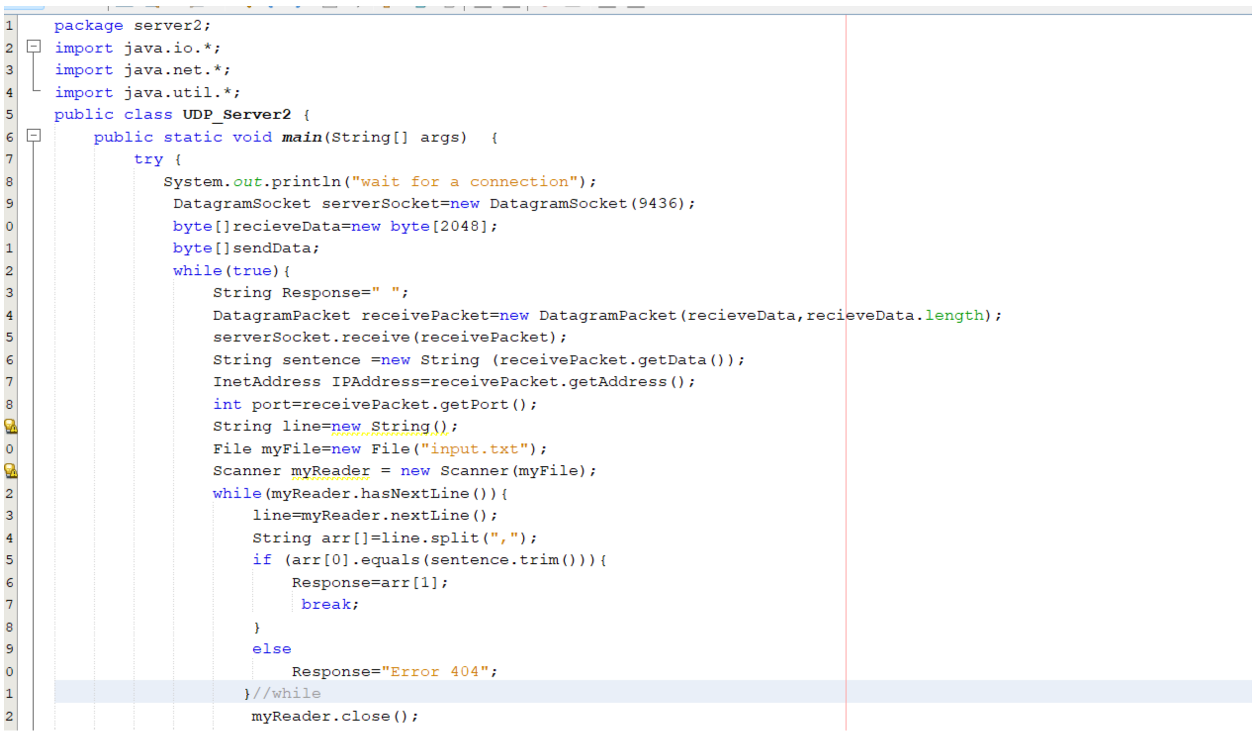
```java
package client2;
import java.io.*;
import java.net.*;
public class UDP Client2 {


    public static void main(String[] args) throws Exception {
        while(true){
        System.out.println("Enter Course Number : ");

        BufferedReader inFromUser= new BufferedReader(new InputStreamReader(System.in));
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress=InetAddress.getByName("127.0.0.1");
        byte[] sendData=new byte[2048];
        byte[]recieveData=new byte[2048];
        String sentense= inFromUser.readLine();
        sendData=sentense.getBytes();
        DatagramPacket sendPacket=new DatagramPacket(sendData,sendData.length,IPAddress ,9436);
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
        clientSocket.receive(receivePacket);
        String OutputSentense=new String (receivePacket.getData());
        System.out.println("FromServer : "+ OutputSentense);
        clientSocket.close();


        }

    }

}
```

- **UDP_CLIENT_CODE_DESCRIPTION :**

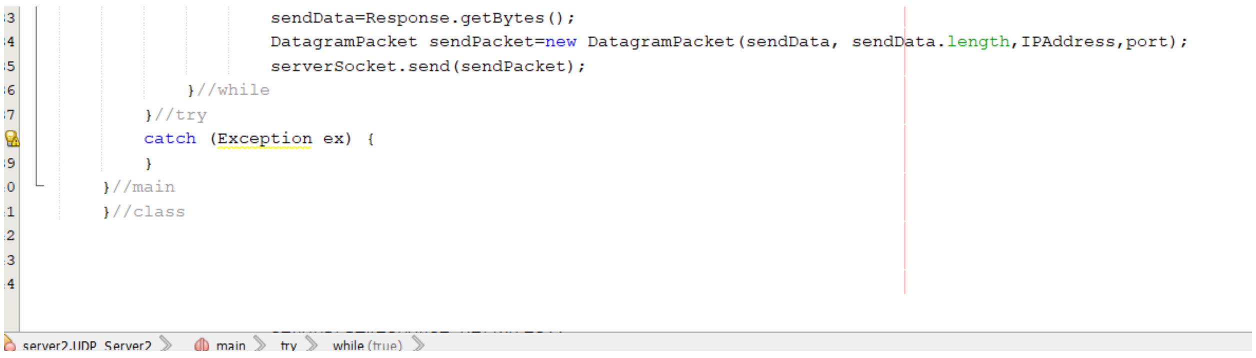
1. Input stream from keyboard (system.in) to read string (course id)
2. Convert it into array of bytes
3. Create socket with type datagram and the port# is default by os
4. Internet address for the server which we will connect with →127.0.0.1 in case of localhost or actual op address if we use different machine
5. Buffer for receive data (array of bytes) but no need of buffer for sending data.
6. Create send packet which contains sendData, length, server address and port# .
7. Send packet by client socket
8. Datagram packet to receive data we give the constructer receive data and maximum length .
9. Client socket waits to receive response packet from server

10. After receiving response packet get only the data and put it in string then print it into monitor
11. Close client socket

➢ **UDP Server Code**



```java
package server2;
import java.io.*;
import java.net.*;
import java.util.*;
public class UDP_Server2 {
    public static void main(String[] args)  {
        try {
           System.out.println("wait for a connection");
            DatagramSocket serverSocket=new DatagramSocket(9436);
            byte[]recieveData=new byte[2048];
            byte[]sendData;
            while(true){
                String Response=" ";
                DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
                serverSocket.receive(receivePacket);
                String sentence =new String (receivePacket.getData());
                InetAddress IPAddress=receivePacket.getAddress();
                int port=receivePacket.getPort();
                String line=new String();
                File myFile=new File("input.txt");
                Scanner myReader = new Scanner(myFile);
                while(myReader.hasNextLine()){
                    line=myReader.nextLine();
                    String arr[]=line.split(",");
                    if (arr[0].equals(sentence.trim())){
                        Response=arr[1];
                         break;
                    }
                    else
                        Response="Error 404";
                   }//while
                    myReader.close();
```



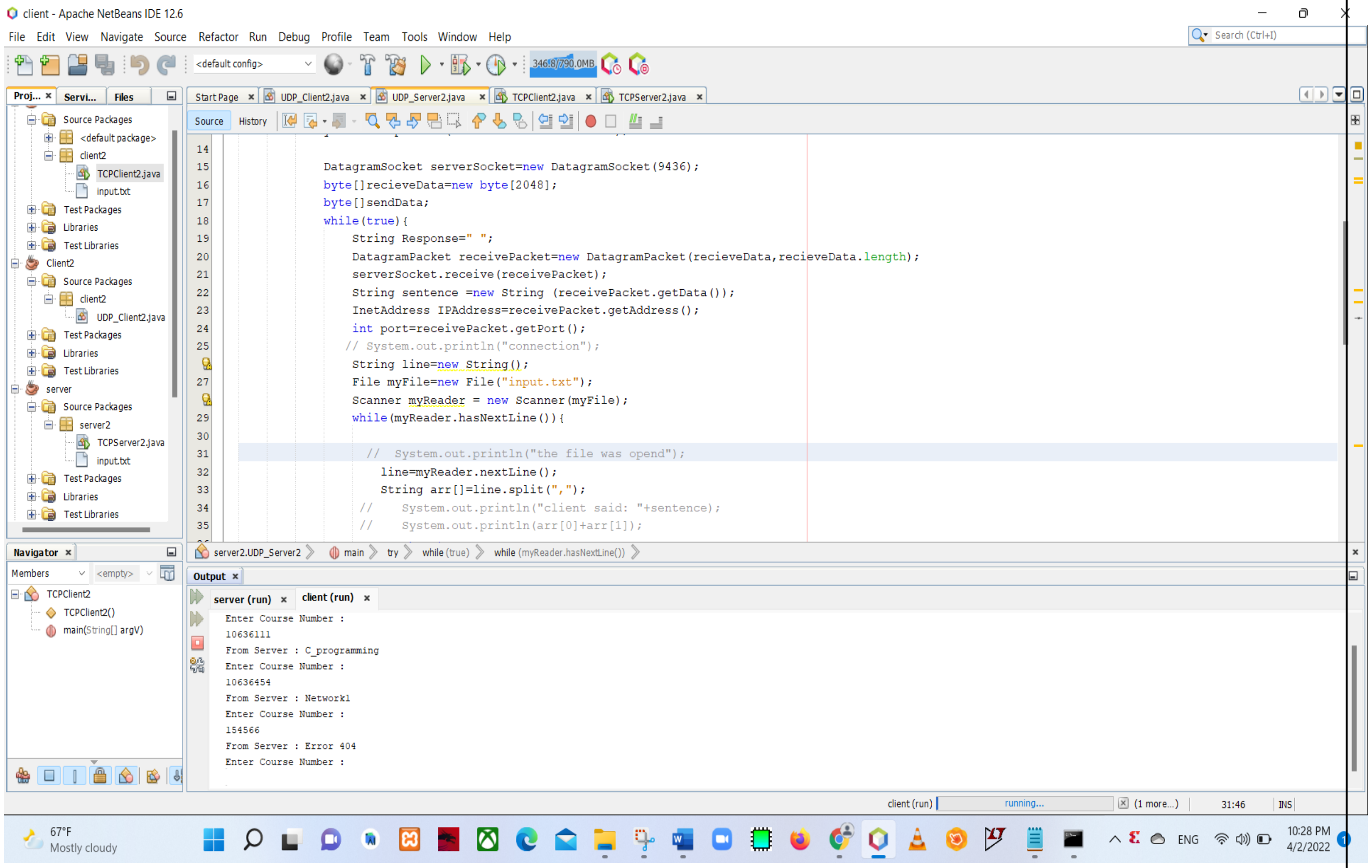
```java
                    sendData=Response.getBytes();
                    DatagramPacket sendPacket=new DatagramPacket(sendData, sendData.length,IPAddress,port);
                    serverSocket.send(sendPacket);
            }//while
        }//try
        catch (Exception ex) {
        }
    }//main
    }//class
```

➢ **UDP_SERVER_CODE_DESCRIPTION :**

1. Create Server socket its type is datagram and give it port number
2. Array of bytes for send & receive data(it need buffer)
3. While loop to get requests and give responses to many clients
4. Create Datagram packet to receive data
5. Server socket waits for a receive packet
6. Get data from packet also get port# and ip address from receive packet

7. Search on the file according to the received data and get the course name corresponding to each course id.
8. Put response on string and convert it into bytes
9. But data into packet with data length ,src_ip and src_port#.
10. Send packet to server.

➢ **UDP_Output :**



```java
        DatagramSocket serverSocket=new DatagramSocket(9436);
        byte[]recieveData=new byte[2048];
        byte[]sendData;
        while(true){
            String Response=" ";
            DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
            serverSocket.receive(receivePacket);
            String sentence =new String (receivePacket.getData());
            InetAddress IPAddress=receivePacket.getAddress();
            int port=receivePacket.getPort();
          // System.out.println("connection");
            String line=new String();
            File myFile=new File("input.txt");
            Scanner myReader = new Scanner(myFile);
            while(myReader.hasNextLine()){

              //  System.out.println("the file was opend");
                line=myReader.nextLine();
                String arr[]=line.split(",");
             //    System.out.println("client said: "+sentence);
             //    System.out.println(arr[0]+arr[1]);
```

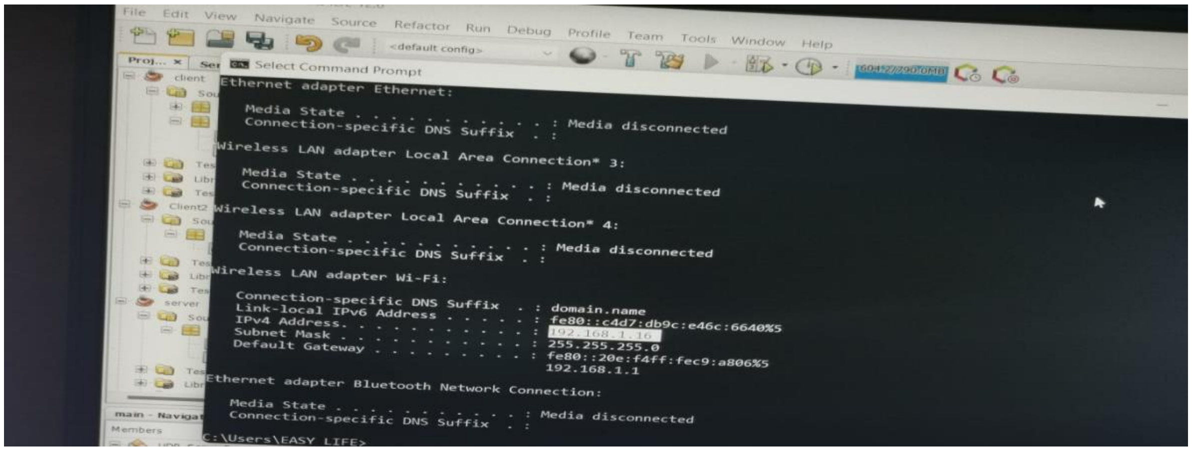
```
Enter Course Number :
10636111
From Server : C_programming
Enter Course Number :
10636454
From Server : Network1
Enter Course Number :
154566
From Server : Error 404
Enter Course Number :
```

TCP & UDP On Different Devices :

➢ **Notes:**

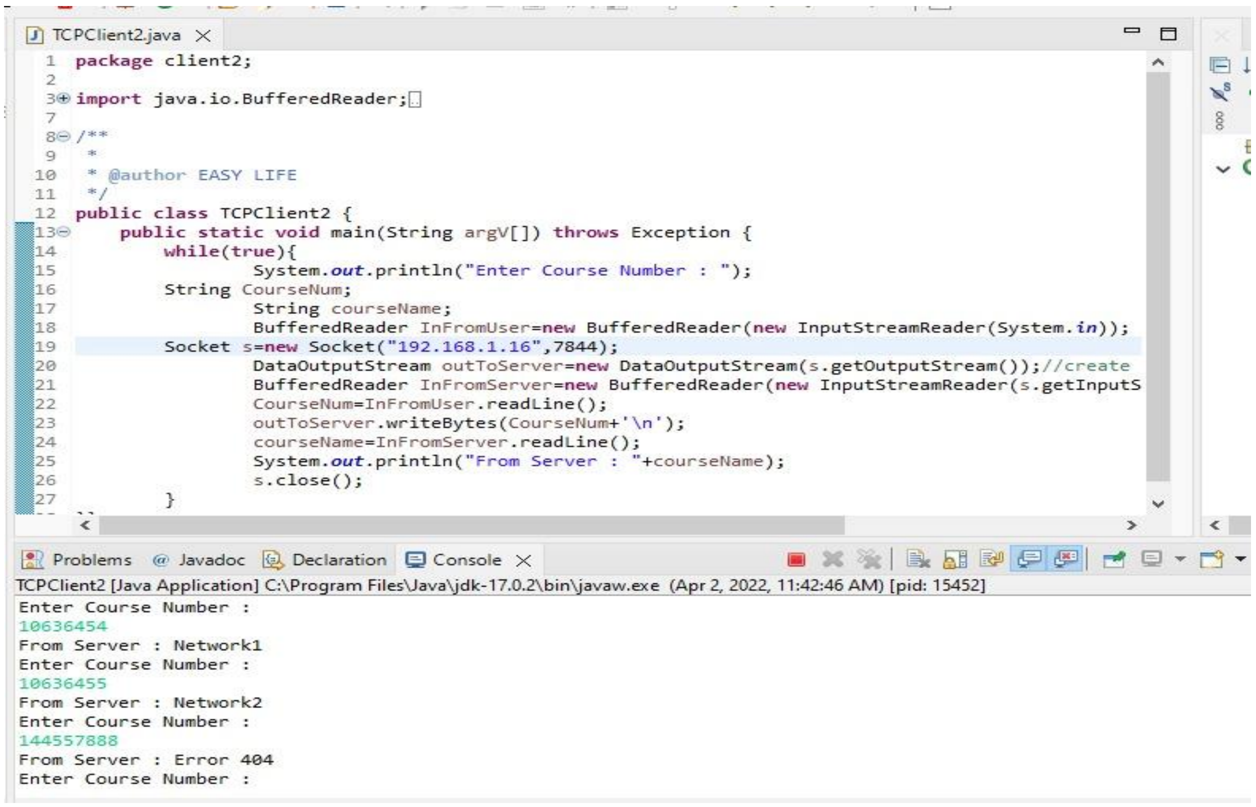
- Putting Server code in device and client code on the other device
- Getting Ip of the other device from ipconfig in cmd
- Give it to the client socket

Ip of the second device as shown below



```
Ethernet adapter Ethernet:

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

Wireless LAN adapter Local Area Connection* 3:

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

Wireless LAN adapter Local Area Connection* 4:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : domain.name
   Link-local IPv6 Address . . . . . : fe80::c4d7:db9c:e46c:6640%5
   IPv4 Address. . . . . . . . . . . : 192.168.1.16
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::20e:f4ff:fec9:a806%5
                                       192.168.1.1

Ethernet adapter Bluetooth Network Connection:

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

C:\Users\EASY LIFE>
```

- **TCP on different devices pictures**
  - Client code on the 1st device (using eclipse)



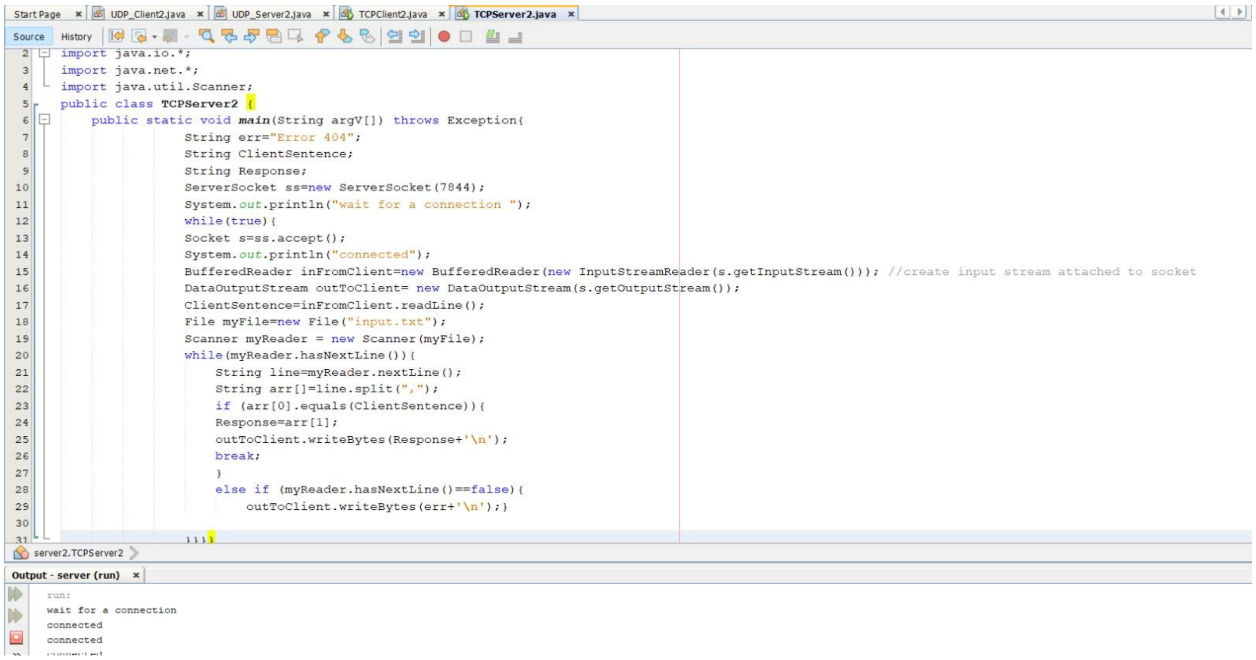
TCPClient2.java

```java
package client2;

import java.io.BufferedReader;
/**
 *
 * @author EASY LIFE
 */
public class TCPClient2 {
    public static void main(String argV[]) throws Exception {
        while(true){
                System.out.println("Enter Course Number : ");
        String CourseNum;
                String courseName;
                BufferedReader InFromUser=new BufferedReader(new InputStreamReader(System.in));
        Socket s=new Socket("192.168.1.16",7844);
                DataOutputStream outToServer=new DataOutputStream(s.getOutputStream());//create
                BufferedReader InFromServer=new BufferedReader(new InputStreamReader(s.getInputS
                CourseNum=InFromUser.readLine();
                outToServer.writeBytes(CourseNum+'\n');
                courseName=InFromServer.readLine();
                System.out.println("From Server : "+courseName);
                s.close();
        }
```

TCPClient2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Apr 2, 2022, 11:42:46 AM) [pid: 15452]

```
Enter Course Number :
10636454
From Server : Network1
Enter Course Number :
10636455
From Server : Network2
Enter Course Number :
144557888
From Server : Error 404
Enter Course Number :
```

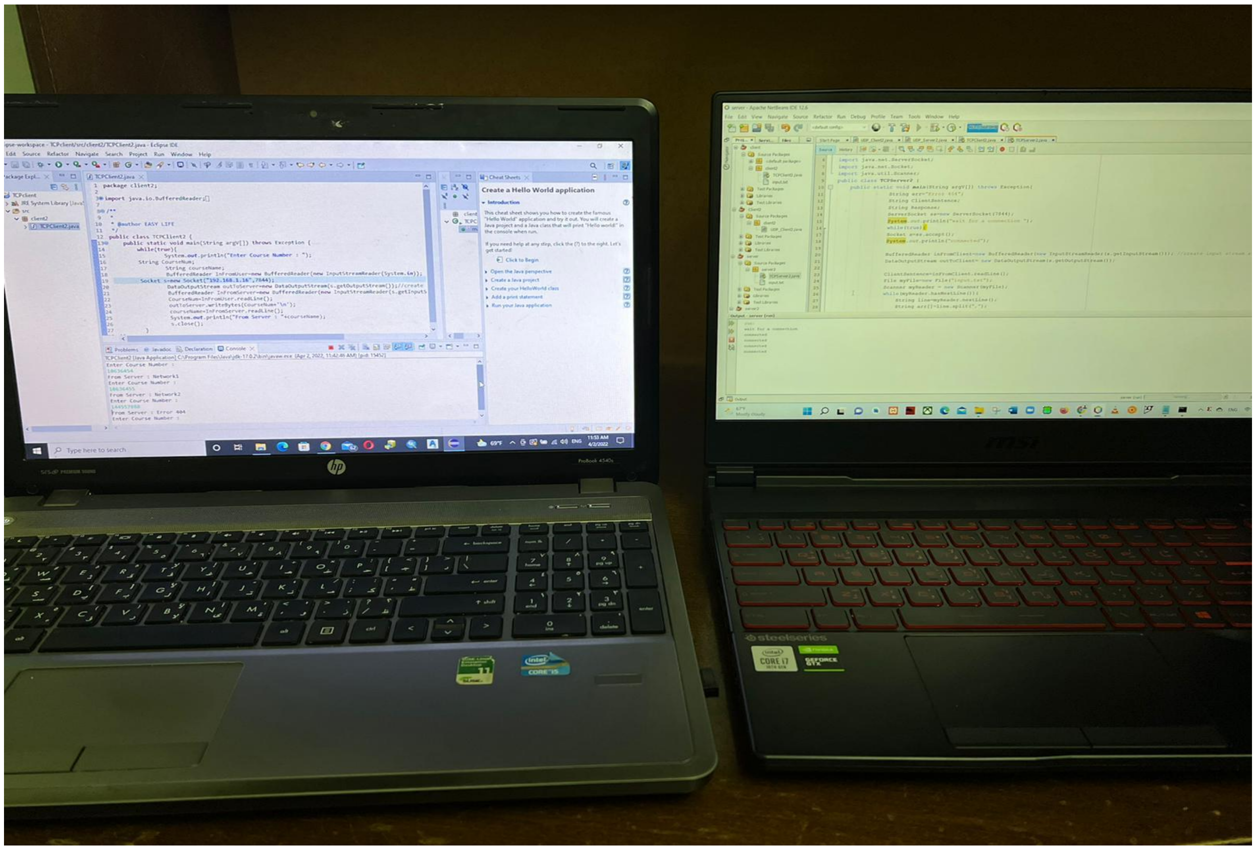
- Server code on the 2nd device (using netbeans)



```java
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class TCPServer2 {
    public static void main(String argV[]) throws Exception{
                String err="Error 404";
                String ClientSentence;
                String Response;
                ServerSocket ss=new ServerSocket(7844);
                System.out.println("wait for a connection ");
                while(true){
                Socket s=ss.accept();
                System.out.println("connected");
                BufferedReader inFromClient=new BufferedReader(new InputStreamReader(s.getInputStream())); //create input stream attached to socket
                DataOutputStream outToClient= new DataOutputStream(s.getOutputStream());
                ClientSentence=inFromClient.readLine();
                File myFile=new File("input.txt");
                Scanner myReader = new Scanner(myFile);
                while(myReader.hasNextLine()){
                    String line=myReader.nextLine();
                    String arr[]=line.split(",");
                    if (arr[0].equals(ClientSentence)){
                    Response=arr[1];
                    outToClient.writeBytes(Response+'\n');
                    break;
                    }
                    else if (myReader.hasNextLine()==false){
                        outToClient.writeBytes(err+'\n');}

                }}}}
```

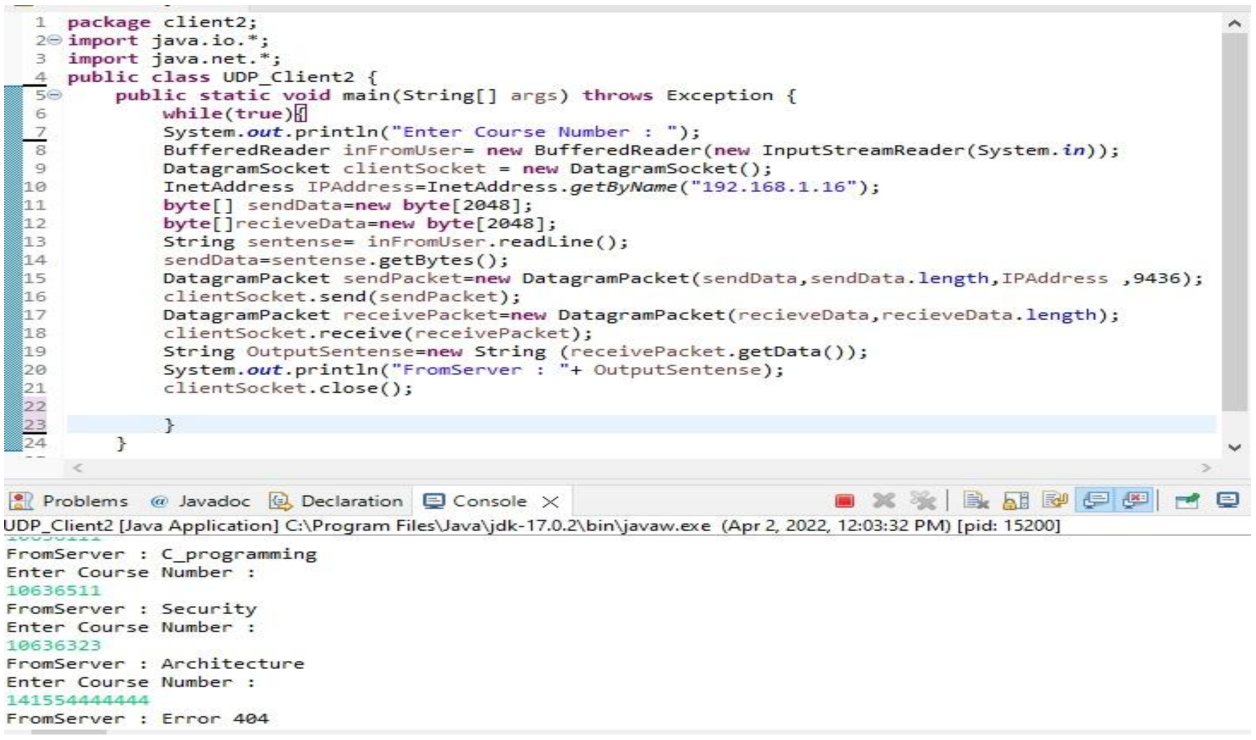
```
run:
wait for a connection
connected
connected
```

- PICTURE FOR BOTH DEVICES WORKING TOGETHER (TCP)



- **UDP on different devices pictures**

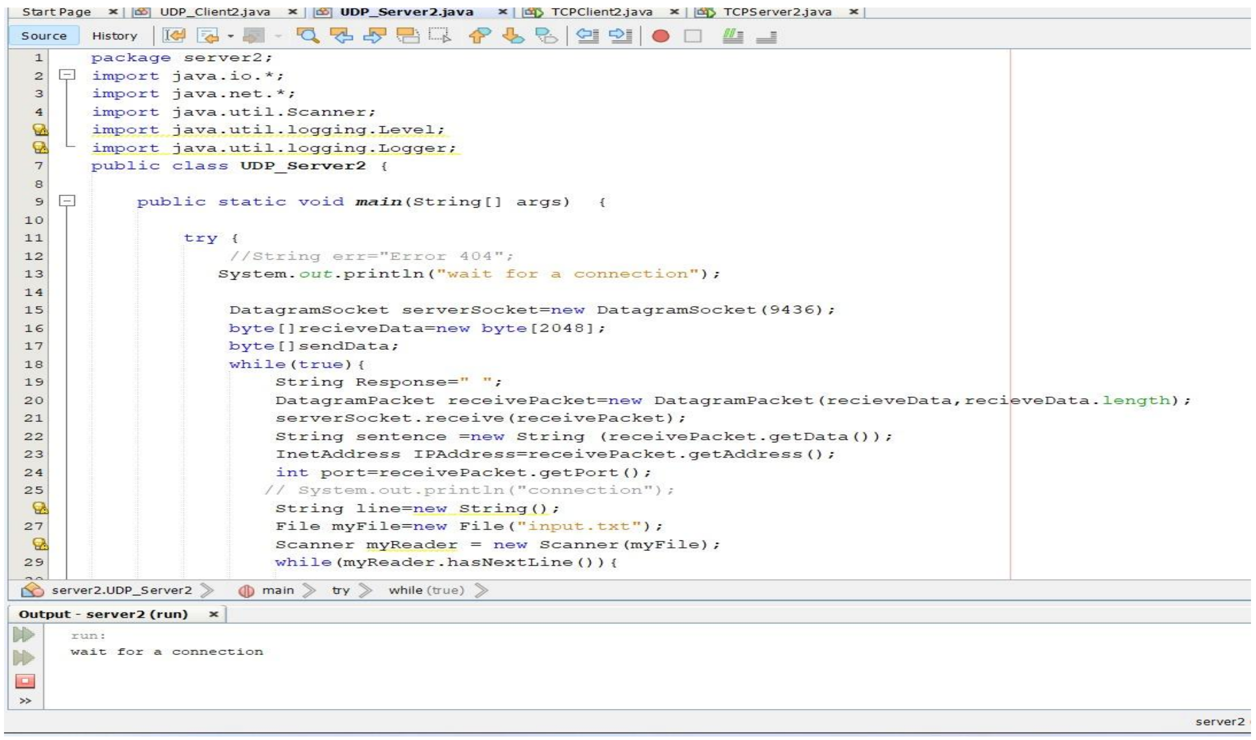
    - Client code on the 1st device (using eclipse)



```java
package client2;
import java.io.*;
import java.net.*;
public class UDP_Client2 {
    public static void main(String[] args) throws Exception {
        while(true){
        System.out.println("Enter Course Number : ");
        BufferedReader inFromUser= new BufferedReader(new InputStreamReader(System.in));
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress=InetAddress.getByName("192.168.1.16");
        byte[] sendData=new byte[2048];
        byte[]recieveData=new byte[2048];
        String sentense= inFromUser.readLine();
        sendData=sentense.getBytes();
        DatagramPacket sendPacket=new DatagramPacket(sendData,sendData.length,IPAddress ,9436);
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
        clientSocket.receive(receivePacket);
        String OutputSentense=new String (receivePacket.getData());
        System.out.println("FromServer : "+ OutputSentense);
        clientSocket.close();

        }
    }
```

```
UDP_Client2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Apr 2, 2022, 12:03:32 PM) [pid: 15200]
FromServer : C_programming
Enter Course Number :
10636511
FromServer : Security
Enter Course Number :
10636323
FromServer : Architecture
Enter Course Number :
141554444444
FromServer : Error 404
```

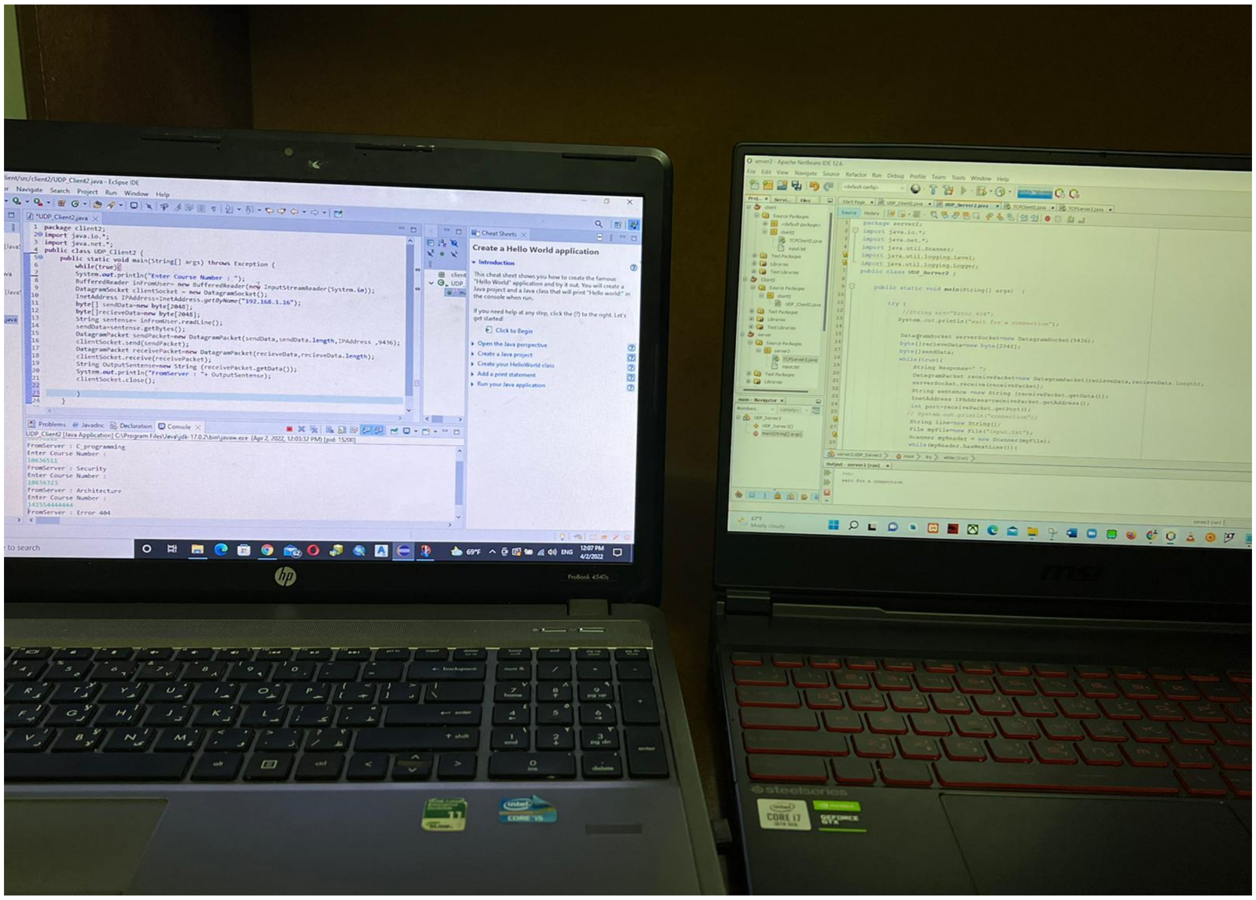
    - Server code on the 2nd device (using netbeans)



```java
package server2;
import java.io.*;
import java.net.*;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
public class UDP_Server2 {

    public static void main(String[] args)  {

        try {
            //String err="Error 404";
           System.out.println("wait for a connection");

            DatagramSocket serverSocket=new DatagramSocket(9436);
            byte[]recieveData=new byte[2048];
            byte[]sendData;
            while(true){
                String Response=" ";
                DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
                serverSocket.receive(receivePacket);
                String sentence =new String (receivePacket.getData());
                InetAddress IPAddress=receivePacket.getAddress();
                int port=receivePacket.getPort();
              // System.out.println("connection");
                String line=new String();
                File myFile=new File("input.txt");
                Scanner myReader = new Scanner(myFile);
                while(myReader.hasNextLine()){
```

```
run:
wait for a connection
```

- PICTURE FOR BOTH DEVICES WORKING TOGETHER (UDP)



```java
package client2;
import java.io.*;
import java.net.*;
public class UDP_Client2 {
    public static void main(String[] args) throws Exception {
        while(true){
        System.out.println("Enter Course Number : ");
        BufferedReader inFromUser= new BufferedReader(new InputStreamReader(System.in));
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress=InetAddress.getByName("192.168.1.16");
        byte[] sendData=new byte[2048];
        byte[]recieveData=new byte[2048];
        String sentense= inFromUser.readLine();
        sendData=sentense.getBytes();
        DatagramPacket sendPacket=new DatagramPacket(sendData,sendData.length,IPAddress ,9436);
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
        clientSocket.receive(receivePacket);
        String OutputSentense=new String (receivePacket.getData());
        System.out.println("FromServer : "+ OutputSentense);
        clientSocket.close();

        }
    }
```

```
UDP_Client2 [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Apr 2, 2022, 12:03:32 PM) [pid: 15200]
FromServer : C_programming
Enter Course Number :
10636511
FromServer : Security
Enter Course Number :
10636323
FromServer : Architecture
Enter Course Number :
141554444444
FromServer : Error 404
```

```java
package server2;
import java.io.*;
import java.net.*;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
public class UDP_Server2 {

    public static void main(String[] args)  {

        try {
            //String err="Error 404";
           System.out.println("wait for a connection");

           DatagramSocket serverSocket=new DatagramSocket(9436);
           byte[]recieveData=new byte[2048];
           byte[]sendData;
           while(true){
               String Response=" ";
               DatagramPacket receivePacket=new DatagramPacket(recieveData,recieveData.length);
               serverSocket.receive(receivePacket);
               String sentence =new String (receivePacket.getData());
               InetAddress IPAddress=receivePacket.getAddress();
               int port=receivePacket.getPort();
              // System.out.println("connection");
               String line=new String();
               File myFile=new File("input.txt");
               Scanner myReader = new Scanner(myFile);
               while(myReader.hasNextLine()){
```

```
run:
wait for a connection
```