ASSIGNMENT NO : A7 - TCP CALCULATOR

Title: Lab Assignment on Unit V: (Mandatory Assignment)
Write a program of Calculator (Arithmetic) using TCP socket for wired network.

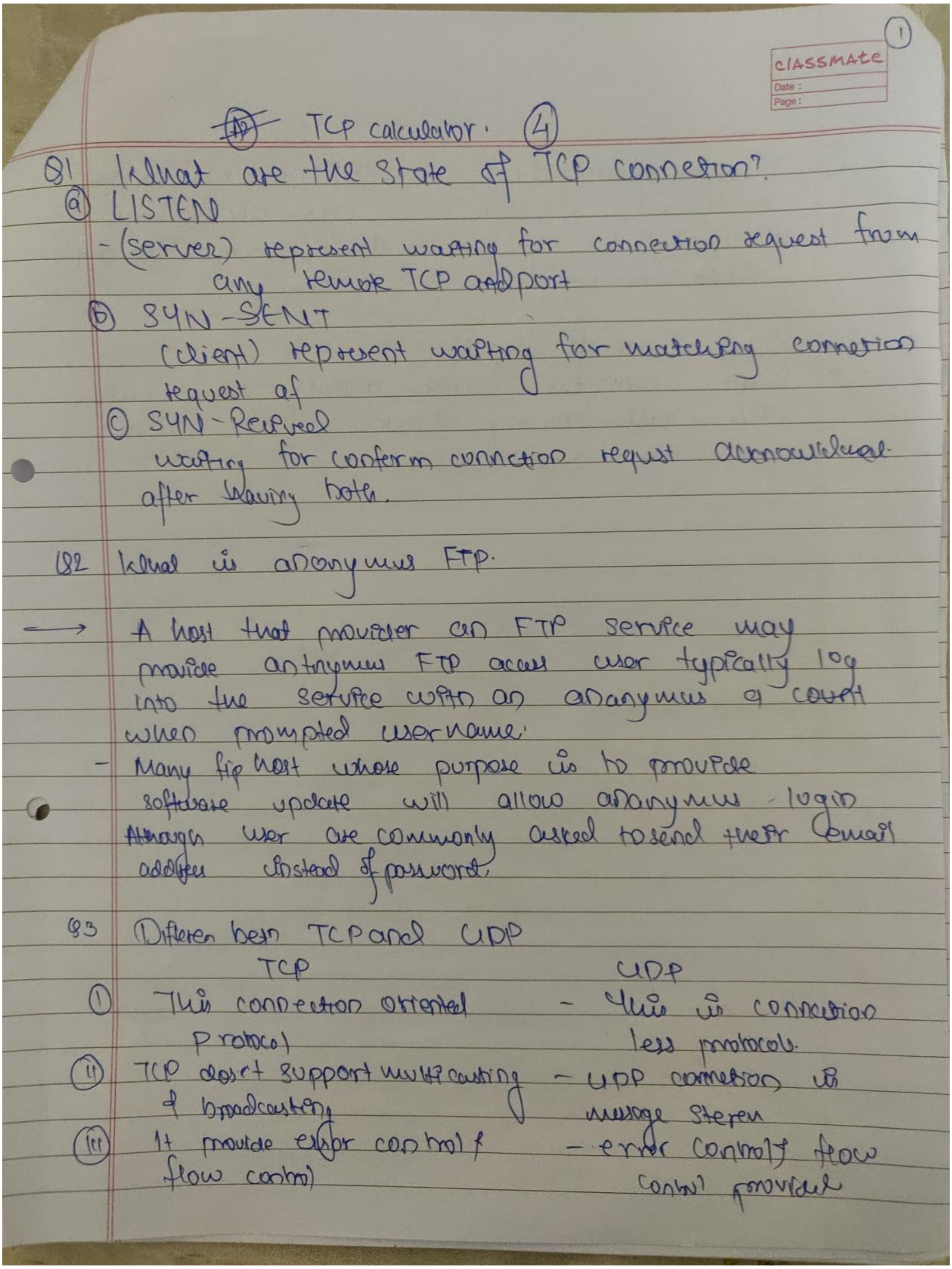
CODE :

1. CalClient.java

```
import java.io.DataInputStream;
import
java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket; import
java.util.Scanner; public class
CalClient {

    public static void main(String[] args) throws IOException {
        InetAddress ip = InetAddress.getLocalHost();
        int port = 4444;
        Scanner sc = new Scanner(System.in);

        // Step 1: Open the socket connection.
        Socket s = new Socket(ip, port);

        // Step 2: Communication-get the input and output stream
        DataInputStream dis = new DataInputStream(s.getInputStream());
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        while (true) {
            // Enter the equation in the form//
            "operand1 operation operand2"
            System.out.print("Enter the equation in the form: ");
            System.out.println("'operand operator operand'");

            String inp = sc.nextLine();

            if (inp.equals("bye"))
                break;

            // send the equation to server
            dos.writeUTF(inp);

            // wait till request is processed and sent back to client
            String ans = dis.readUTF();
            System.out.println("Answer = " + ans);
        }
    }
}

/* OUTPUT
Enter the equation in the form: 'operand operator operand'
78 + 23
Answer = 101
Enter the equation in the form: 'operand operator operand'
15 / 3
Answer = 5
Enter the equation in the form: 'operand operator operand'
43 - 24
Answer = 19
Enter the equation in the form: 'operand operator operand'
12 * 4
Answer = 48
Enter the equation in the form: 'operand operator operand'
bye */
```

2. CalServer.java

```
import java.io.DataInputStream; import
java.io.DataOutputStream; import java.io.IOException; import
java.net.ServerSocket; import java.net.Socket; import
java.net.SocketException; import java.util.StringTokenizer;
public class CalServer { public static void main(String
args[]) throws IOException {

        // Step 1: Establish the socket connection.
        ServerSocket ss = new ServerSocket(4444);
        Socket s = ss.accept();

        // Step 2: Processing the request.
        DataInputStream dis = new DataInputStream(s.getInputStream());
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        while (true) {
            // wait for input String
            input = dis.readUTF();

            if (input.equals("bye")) {
                break;
            }

            System.out.println("Equation received : " + input);
            int result;

            // Use StringTokenizer to break the equation into operand and
            // operation
            StringTokenizer st = new StringTokenizer(input);

            int oprnd1 = Integer.parseInt(st.nextToken());
            String operation = st.nextToken(); int oprnd2
            = Integer.parseInt(st.nextToken()); // perform
            the required operation.
            if (operation.equals("+")) {
                result = oprnd1 + oprnd2;
            }

            else if (operation.equals("-")) {
                result = oprnd1 - oprnd2;
            } else if (operation.equals("*")) {
                result = oprnd1 * oprnd2;
            } else { result = oprnd1 /
                oprnd2;
            }
            System.out.println("Sending the result... [" + oprnd1 + operation + oprnd2
+ " = " + result + "]");

            // send the result back to the client.
            dos.writeUTF(Integer.toString(result));
        }
    }
}



/*
*       OUTPUT
 *
C:\Users\dell\eclipse-workspace\CNL A7 TCP Calculator\src>javac CalServer.java

C:\Users\dell\eclipse-workspace\CNL A7 TCP Calculator\src>java CalServer
Equation received : 78 + 23
Sending the result... [78+23 = 101]
Equation received : 15 / 3
Sending the result... [15/3 = 5]
Equation received : 43 - 24
Sending the result... [43-24 = 19]
Equation received : 12 * 4
Sending the result... [12*4 = 48]
Exception in thread "main" java.net.SocketException: Connection reset at
        java.base/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:323) at
        java.base/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:350) at
        java.base/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:803) at
        java.base/java.net.Socket$SocketInputStream.read(Socket.java:982) at
        java.base/java.net.Socket$SocketInputStream.read(Socket.java:977) at
java.base/java.io.DataInputStream.readUnsignedShort(DataInputStream.java:341)
        at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:593)
        at java.base/java.io.DataInputStream.readUTF(DataInputStream.java:568)
        at CalServer.main(CalServer.java:22)

C:\Users\dell\eclipse-workspace\CNL A7 TCP Calculator\src>
 * */
```



TCP calculator. 4

Q1 What are the state of TCP connetion?

a) LISTEN

- (server) represent waiting for connection request from any remote TCP and port

b) SYN-SENT

(client) represent waiting for matching connetion request af

c) SYN-Received

waiting for conferm connection requst acknowldegal after having both.

Q2 What is anonymus FTP.

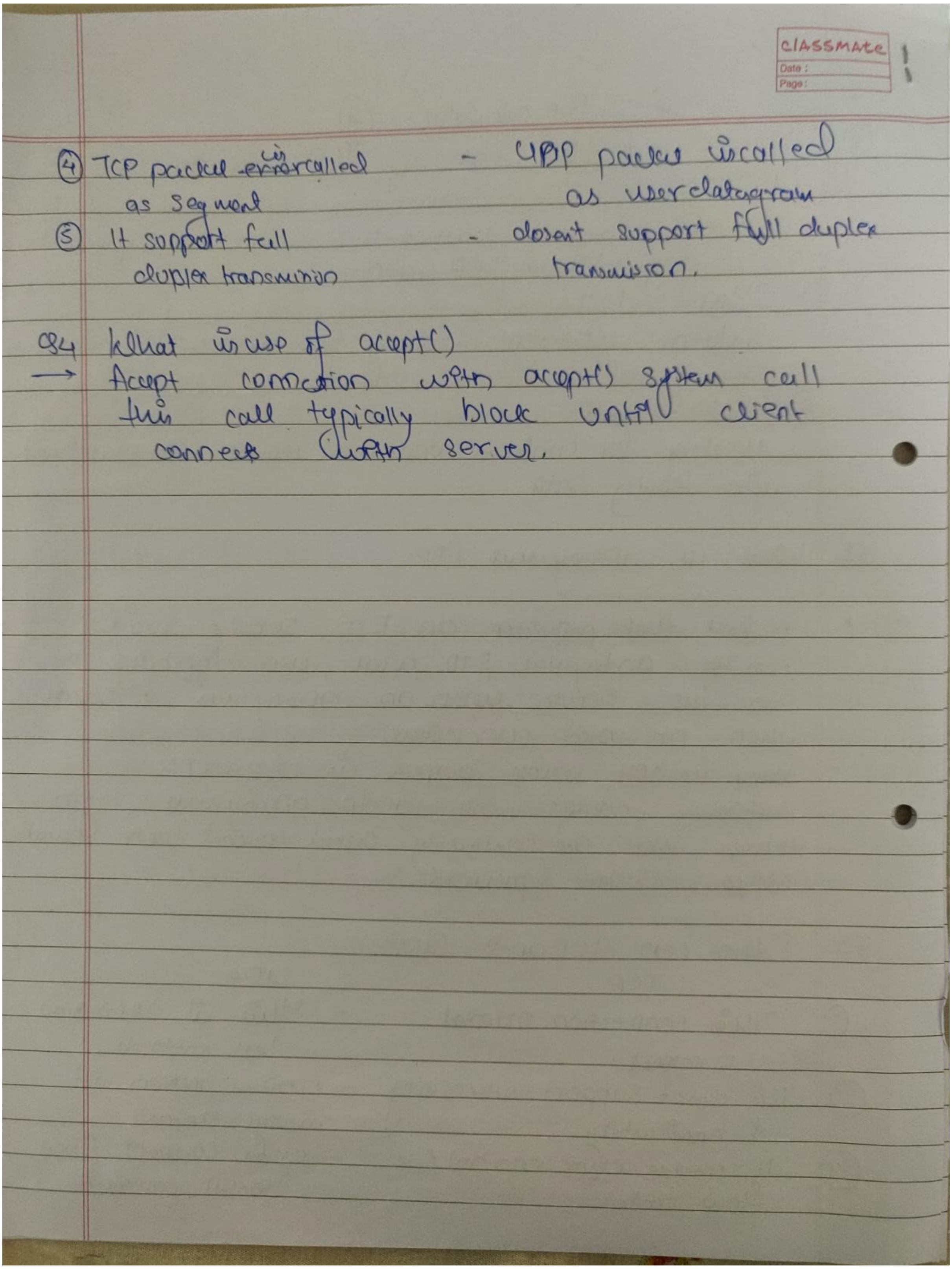
→ A host that provider an FTP service may provide antnymus FTP acess user typically log into the service with an ananymus a count when prompted username.

- Many ftp host whose purpose is to provide software update will allow ananymus login Although user are commonly asked to send their email addres instead of password.

Q3 Differen betn TCP and UDP

	TCP	UDP
(i)	This connection oriented protocol	- This is connection less protocols.
(ii)	TCP dosn't support multicasting & broadcasting	- UDP connetion is message steren
(iii)	It provide error control & flow control	- error control & flow control provided

Pushkaraj Chavan TEC 58



classmate
Date :
Page :

④ TCP packet ~~erro~~ is called as segment	- UDP packet is called as user datagram
⑤ It support full duplex transmision	- dosen't support full duplex transmisson.

Q4 What is use of accept()

→ Accept connection with accept() system call this call typically block until client connects with server.

Pushkaraj Chavan TEC 58

Name : Pushkaraj Chavan.

Class : TE

Div : C

Roll No. : 58