VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

COMPUTER NETWORKS

LAB ASSESSMENT - 1

SUBMITTED BY:-

Vaidya Urmila Suman

21MIS1098

A) BASIC NETWORK COMMANDS

AIM:- To perform basic network commands

1) IPCONFIG

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\student> ipconfig
Windows IP Configuration

Ethernet adapter Ethernet:
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.16.15.62
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.15.2
PS C:\Users\student>
```

2) NSLOOKUP

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\student> nslookup
Default Server:  vitccdns
Address:  172.16.1.11
>
```

3) HOSTNAME

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\student> hostname
AB1311SCOPE62
PS C:\Users\student>
```

4)PING

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> PING

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.

PS C:\Users\student>
```

5) TRACERT

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> TRACERT

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
PS C:\Users\student>
```

1) NETSTAT

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> NETSTAT

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    172.16.15.62:49681     20.198.119.84:https    ESTABLISHED
  TCP    172.16.15.62:49755     13.107.5.88:https      ESTABLISHED
  TCP    172.16.15.62:49788     se-in-f188:5228        ESTABLISHED
  TCP    172.16.15.62:49868     13.107.21.200:https    ESTABLISHED
  TCP    172.16.15.62:49874     204.79.197.222:https   TIME_WAIT
  TCP    172.16.15.62:49881     13.107.136.254:https   ESTABLISHED
  TCP    172.16.15.62:49882     13.107.4.254:https     TIME_WAIT
  TCP    172.16.15.62:49883     144.2.15.25:https      ESTABLISHED
  TCP    172.16.15.62:49884     52.98.86.162:https     ESTABLISHED
  TCP    172.16.15.62:49885     13.107.4.254:https     ESTABLISHED
  TCP    172.16.15.62:49886     117.18.232.200:https   ESTABLISHED
  TCP    172.16.15.62:49887     152.199.43.62:https    ESTABLISHED
  TCP    172.16.15.62:49888     204.79.197.222:https   ESTABLISHED
PS C:\Users\student>
```

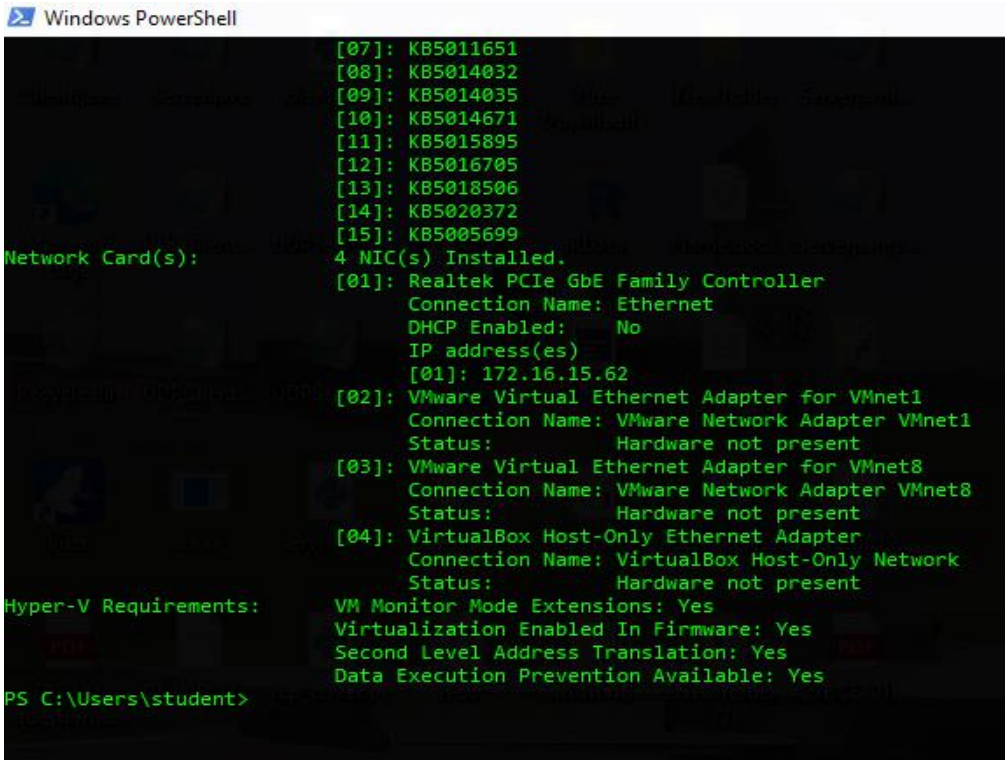
2) ARP

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> ARP

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  -v            Displays current ARP entries in verbose mode.  All invalid
                entries and entries on the loop-back interface will be shown.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.
PS C:\Users\student>
```

3)SYSTEMINFO

Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> SYSTEMINFO

Host Name:                 AB1311SCOPE62
OS Name:                   Microsoft Windows 10 Pro
OS Version:                10.0.19045 N/A Build 19045
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:          nsrl
Registered Organization:
Product ID:                00331-10000-00001-AA531
Original Install Date:     15-02-2022, 04:38:25 PM
System Boot Time:          18-05-2023, 10:09:14 AM
System Manufacturer:       HP
System Model:              HP 280 G2 MT
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 94 Stepping 3 GenuineIntel ~3192 Mhz
BIOS Version:              AMI A0.52, 02-12-2019
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume2
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory:     8,047 MB
Available Physical Memory: 4,549 MB
Virtual Memory: Max Size:  10,351 MB
Virtual Memory: Available: 6,692 MB
Virtual Memory: In Use:    3,659 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\AB1311SCOPE62
Hotfix(s):                 15 Hotfix(s) Installed.
                           [01]: KB5022502
                           [02]: KB5003791
                           [03]: KB5012170
                           [04]: KB5015684
                           [05]: KB5022834
                           [06]: KB5011352
                           [07]: KB5011651
                           [08]: KB5014032
                           [09]: KB5014035
                           [10]: KB5014671
                           [11]: KB5015895
```



B) Write a program to implement a simple message transfer from client to server process using tcp sockets.

Aim: To write a program for transferring a simple message using TCP socket programming.

Algorithm:

Server side:

1. Create a ServerSocket object and bind it to a port number.
2. Listen for incoming connections using the accept() method.
3. When a client connects:
   3.1. Create an InputStream object to receive data from the client.
   3.2. Create a String to read input from client using readUTF() method.
   3.3. Send a welcome message to the client.
   3.4. Close the streams and socket when the client disconnects
4. Close the ServerSocket object

Client side:

1. Create a Socket object and connect to the server on a specified port number.
2. Create an OutputStream object to send data to the server.
3. Print the welcome message using writeUTF() method.

Close the streams and socket when the user quits the program

S_Class.java

```
import java.io.*;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.Scanner;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;


public class S_Class {

   int pt;

   ServerSocket ss = null;

   Socket socket = null;

   ExecutorService es = null;

   int clientcount = 0;


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

      S_Class sObject = new S_Class(5000);

      sObject.startServer();

   }


   S_Class(int pt) {

      this.pt = pt;

      es = Executors.newFixedThreadPool(5);

   }


   public void startServer() throws IOException {


      ss = new ServerSocket(5000);

      System.out.println("S_Class Started....");

      System.out.println("To break the connection send BYE....");

      while (true) {

         socket = ss.accept();

         clientcount++;

         ServerThread st = new ServerThread(socket, clientcount, this);

         es.execute(st);

      }


   }


   private static class ServerThread implements Runnable {
```

```
S_Class server = null;
Socket client = null;
BufferedReader s1;
PrintStream s2;
Scanner sc = new Scanner(System.in);
int id;
String s;

ServerThread(Socket client, int count, S_Class server) throws IOException {

    this.client = client;
    this.server = server;
    this.id = count;
    System.out.println("Connection established with client " + id);

    s1 = new BufferedReader(new InputStreamReader(client.getInputStream()));
    s2 = new PrintStream(client.getOutputStream());

}

@Override
public void run() {
    int x = 1;
    try {
        while (true) {
            s = s1.readLine();
            System.out.print("Client(" + id + ") : " + s + "\n");
            System.out.print("S_Class : ");
            s = sc.nextLine();
            if (s.equalsIgnoreCase("bye")) {
                s2.println("BYE");
                x = 0;
                System.out.println("Connection Ended....");
                break;
            }
            s2.println(s);
        }

        s1.close();
        client.close();
```

```
            s2.close();
            if (x == 0) {
                System.out.println("*****Closing*****");
                System.exit(0);
            }
        } catch (IOException e) {
            System.out.println("Error : " + e);
        }
    }
  }
}
```

C_Class.java

```
import java.io.*;
import java.net.*;

public class C_Class {

    public static void main(String args[]) throws Exception {
        Socket socket = new Socket("localhost", 5000);
        BufferedReader inputVar = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintStream outVar = new PrintStream(socket.getOutputStream());
        BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
        String str;
        while (true) {
            System.out.print("Client : ");
            str = stdin.readLine();
            outVar.println(str);
            if (str.equalsIgnoreCase("BYE")) {
                System.out.println("Connection Broken.....");
                break;
            }
            str = inputVar.readLine();
            System.out.print("Server : " + str + "\n");

        }
        socket.close();
```

```
        inputVar.close();
        outVar.close();
        stdin.close();
      }

    }
```

OUTPUT:-

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> notepad
PS C:\Users\student\21mis1098> notepad
PS C:\Users\student\21mis1098> notepad
PS C:\Users\student\21mis1098> javac  S_Class.java
PS C:\Users\student\21mis1098> java  S_Class.java
S_Class Started....
To break the connection send BYE....
Connection established with client 1
Client(1) : hii
S_Class : hello
Client(1) : this is urmila
S_Class : this is sanjana, nice to meet u
Client(1) : nice to meet u too
S_Class : bye
Connection Ended....
*****Closing*****
PS C:\Users\student\21mis1098>
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> javac C_Class.java
PS C:\Users\student\21mis1098> java C_Class.java
Client : hii
Server : hello
Client : this is urmila
Server : this is sanjana, nice to meet u
Client : nice to meet u too
Server : BYE
Client :
```

C) Write a tcp socket program to display in client window the sum of random numbers generated by the server.

Aim: To write a TCP socket program, the sum of random numbers generated by the server display in the client.

Algorithm:

1. Create a ServerSocket object and bind it to a port number
2. Listen for incoming connections using the accept() method
3. When a client connects:
   - 3.1. Create an InputStream object to receive data from the client
   - 3.2. Create an OutputStream object to send data to the client
   - 3.3. Create a DataInputStream object to read data from the client
   - 3.4. Create a DataOutputStream object to send data to the client
   - 3.5. Generate a random number between 1 and 10
   - 3.6. Send the number to the client using the DataOutputStream object
   - 3.7. Repeat the following steps until all numbers have been sent:
     - 3.7.1. Generate another random number between 1 and 10
     - 3.7.2. Send the number to the client using the DataOutputStream object
   - 3.8. Close the streams and socket when all numbers have been sent
4. Close the ServerSocket object

CODE:-

Calc_Server.java

```
// Java program to illustrate Server Side Programming

// for Simple Calculator using TCP

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.StringTokenizer;


public class Calc_Server

{

	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...");

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

Calc_Client.java

```
// Java program to illustrate Client Side Programming
// for Simple Calculator using TCP
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class Calc_Client
{
	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"))
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> javac Calc_Server.java
PS C:\Users\student\21mis1098> java Calc_Server.java
Equation received:-5 * 6
Sending the result...
Equation received:-5 + 6
Sending the result...
Equation received:-6 / 2
Sending the result...
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> javac Calc_Client.java
PS C:\Users\student\21mis1098> java Calc_Client.java
Enter the equation in the form: 'operand operator operand'
5 * 6
Answer=30
Enter the equation in the form: 'operand operator operand'
5 + 6
Answer=11
Enter the equation in the form: 'operand operator operand'
6 / 2
Answer=3
Enter the equation in the form: 'operand operator operand'
```

(d) Write a program to implement a chat server and client in Java using TCP sockets.

<u>Aim:</u> To write a program to implement a chat server and client in Java using TCP sockets.

<u>Algorithm:</u>

Server side:

1. Create a ServerSocket object and bind it to a port number
2. Listen for incoming connections using the accept() method
3. When a client connects:
   - 3.1. Create a new thread to handle the client connection 3.2. Store the thread object in a list or map for later access
4. In the thread's run() method:
   - 4.1. Create an InputStream object to receive data from the client
   - 4.2. Create an OutputStream object to send data to the client

4.3. Create a BufferedReader object to read messages from the client
4.4. Create a PrintWriter object to send messages to the client
4.5. Send a welcome message to the client using the PrintWriter object
4.6. Repeat the following steps while the client is connected:
4.6.1. Read a message from the client using the BufferedReader object
4.6.2. Broadcast the message to all clients by iterating through the list of threads and sending the message using each thread's PrintWriter object
4.7. Remove the thread object from the list or map when the client disconnects
4.8. Close the streams and socket

5. Close the ServerSocket object

Client side:

1. Create a Socket object and connect to the server on a specified port number
2. Create an InputStream object to receive data from the server
3. Create an OutputStream object to send data to the server
4. Create a BufferedReader object to read messages from the server 5. Create a PrintWriter object to send messages to the server 6. Repeat the following steps while the connection is open:

6.1. Read a message from the server using the BufferedReader object
6.2. Display the message to the user
6.3. Prompt the user to enter a message
6.4. Send the message to the server using the PrintWriter object

7. Close the streams and socket when the user quits the chat

ServerChat.java

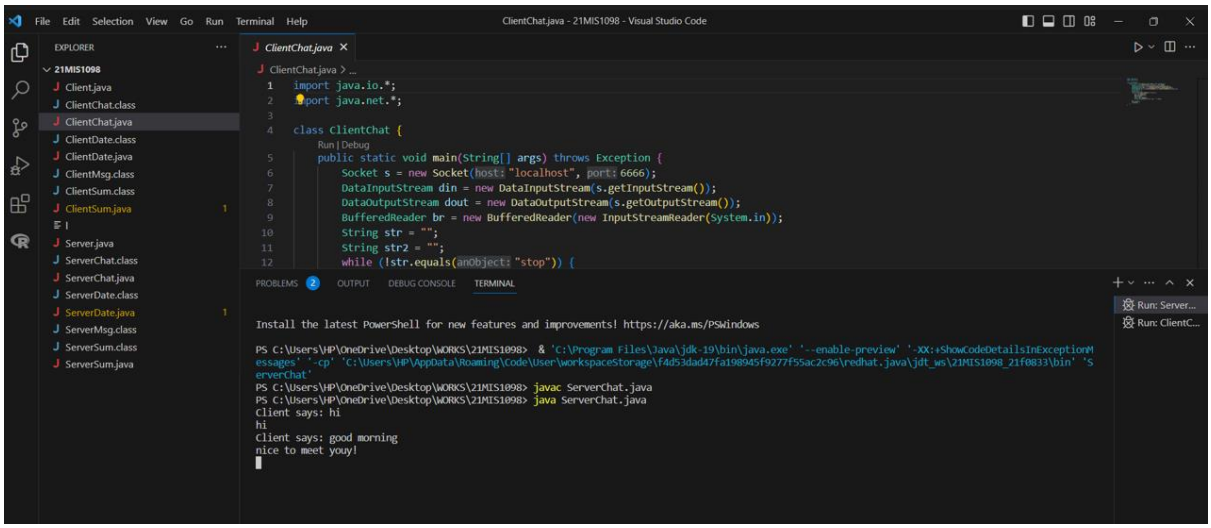
```
import java.io.*;

import java.net.*;

class ServerChat{

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

      ServerSocket ss=new ServerSocket(6666);

      Socket s=ss.accept();

      DataInputStream din=new DataInputStream(s.getInputStream());

      DataOutputStream dout=new DataOutputStream(s.getOutputStream());

      BufferedReader br=new BufferedReader(new

InputStreamReader(System.in));

      String str="";

      String str2="";
```

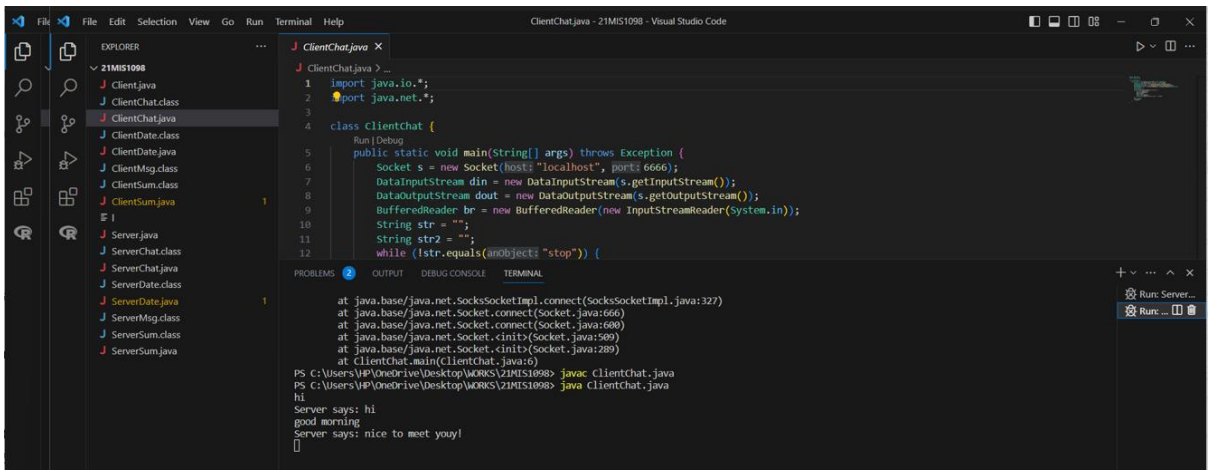
```
        while(!str.equals("stop")){
        str=din.readUTF();
        System.out.println("Client says: "+str);
        str2=br.readLine();
        dout.writeUTF(str2);
        dout.flush();
        }
        din.close();
        s.close();
        ss.close();
    }
}
```

ClientChat.java

```
import java.io.*;
import java.net.*;
class ClientChat{
    public static void main(String[] args) throws Exception{
        Socket s=new Socket("localhost",6666);
        DataInputStream din=new DataInputStream(s.getInputStream());
        DataOutputStream dout=new DataOutputStream(s.getOutputStream());
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
        String str="";
        String str2="";
        while(!str.equals("stop")){
        str=br.readLine();
        dout.writeUTF(str);
        dout.flush();
        str2=din.readUTF();
        System.out.println("Server says: "+str2);
```

```
        }
        dout.close();
        s.close();
    }
}
```





E) Using tcp sockets, write a simple java program to display the current date and time.

Aim: To write a program for displaying current date and time using TCP sockets.

Algorithm:

SERVER SIDE

1. Create a ServerSocket object and bind it to a port number
2. Listen for incoming connections using the accept() method
3. When a client connects:
    3.1. Create an OutputStream object to send data to the client

3.2. Create a PrintWriter object to send messages to the client
3.3. Get the current date and time using the java.util.Date class
3.4. Format the date and time as a string using SimpleDateFormat
3.5. Send the string to the client using the PrintWriter object
3.6. Close the streams and socket

4. Close the ServerSocket object

Client side:

1. Create a Socket object and connect to the server on a specified port number
2. Create an InputStream object to receive data from the server
3. Create a BufferedReader object to read messages from the server
4. Create a PrintWriter object to send messages to the server
5. Send a request message to the server indicating that the client wants the current date and time
6. Read the response message from the server using the BufferedReader object
7. Display the date and time to the user
8. Close the streams and socket when the user is done

Code:-

Dateclient.java

```
import java.io.*;
import java.net.*;
class DateClient
{
   public static void main(String args[]) throws Exception
   {
      Socket soc=new Socket(InetAddress.getLocalHost(),5217);
      BufferedReader in=new BufferedReader(new InputStreamReader(soc.getInputStream()  ));
      System.out.println(in.readLine());
   }
}
```

Dateserver.java

```
import java.net.*;
import java.io.*;
import java.util.*;
class DateServer
{
   public static void main(String args[]) throws Exception
   {
      ServerSocket s=new ServerSocket(5217);
      while(true)
      {
         System.out.println("Waiting For Connection ...");
         Socket soc=s.accept();
```

```
            DataOutputStream out=new DataOutputStream(soc.getOutputStream());
            out.writeBytes("Server Date: " + (new Date()).toString() + "\n");
            out.close();
            soc.close();
        }
    }
}
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> javac dateserver.java
PS C:\Users\student\21mis1098> java dateserver.java
Waiting For Connection ...
Waiting For Connection ...
```

```
Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\student> cd 21mis1098
PS C:\Users\student\21mis1098> javac dateclient.java
PS C:\Users\student\21mis1098> java dateclient.java
Server Date: Thu May 11 12:33:42 IST 2023
PS C:\Users\student\21mis1098>
```