Name : Vaishnavi Shukla

Reg no:- 21MIS1100

Lab :-1

Subject :- CN

(a).Basic Network commands.

1.PingThepingcommand is a networking utility used to test the reachability of a host on an Internet Protocol (IP) network. It can be used to determine whether a specific IP address is accessibleand also measures the round-trip time for messages sent from thelocal hostto a remote host.

2.NetStatNetStatis a networking utility that can be used to display all active network connections and their status. It can be used to identify which applications areusing which ports and can be helpful in troubleshooting networking issues.

3.NbtStatNbtStatis a networking utility in Windows that helps users troubleshoot NetBIOS over TCP/IP problems. It can be used to display a variety of information about the current state of the NetBIOS over TCP/IP protocol on both local and remote computers.

4.IpConfigIP Config is a command-line tool that is used to display the current IP address configuration of a Windows machine. This includes the IP address, subnet mask, anddefault gateway.

5.HostnameIf you know which switch to use with the command,NbtStatcan provide you with the hostnamethat has been assigned to a Windows device, as discussed earlier in this piece.

6.ARPThe Address Resolution Protocol, or ARP, is a networking utility used for mapping network addresses to physical addresses.

7.Nslookup:-Nslookupis a command-line networking tool used for querying Domain Name System (DNS) to obtain domain name or IP address mapping, or other DNS records.

8.TracertThetracertcommand is a Command Prompt command that displays the network packets being sent and received, as well as the number of hops required for them to reach their destination.

9.GetMacThe MAC address is a unique identifier for every network-capabledevice on the internet. The number is assigned during the manufacturing process and is stored in the device’s hardware.

10.Route:The Route networking command is one of the most essential networking commands for Windows administrators.

11.NetDiagNetDiag is a networking diagnostic tool that is included with the Windows operating system

(b) Simple message transfer using TCP Socket Programming.

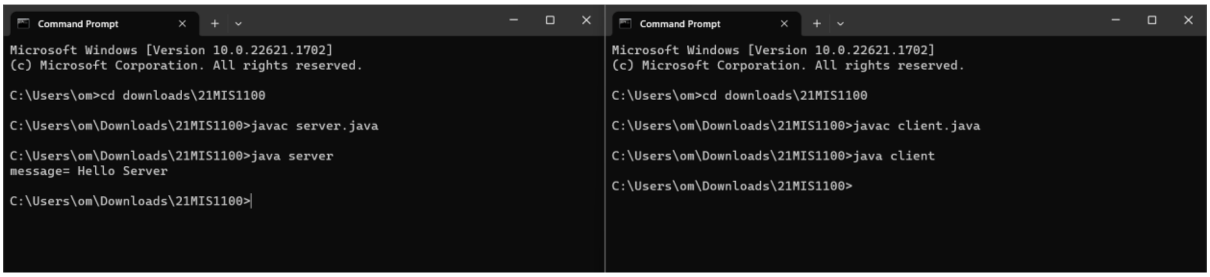
Server:

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

public class server {
    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(6666);
            Socket s = ss.accept();// establishes connection
            DataInputStream dis = new
DataInputStream(s.getInputStream());
            String str = (String) dis.readUTF();
            System.out.println("message= " + str);
            ss.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Client:

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

public class client {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("localhost", 6666);
            DataOutputStream dout = new DataOutputStream(s.getOutputStream());
            dout.writeUTF("Hello Server");
            dout.flush();
            dout.close();
            s.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Output:-



```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac server.java

C:\Users\om\Downloads\21MIS1100>java server
message= Hello Server

C:\Users\om\Downloads\21MIS1100>
```

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac client.java

C:\Users\om\Downloads\21MIS1100>java client

C:\Users\om\Downloads\21MIS1100>
```

(c) Write a TCP socket program, the sum of random numbers generated by the server display in the client.

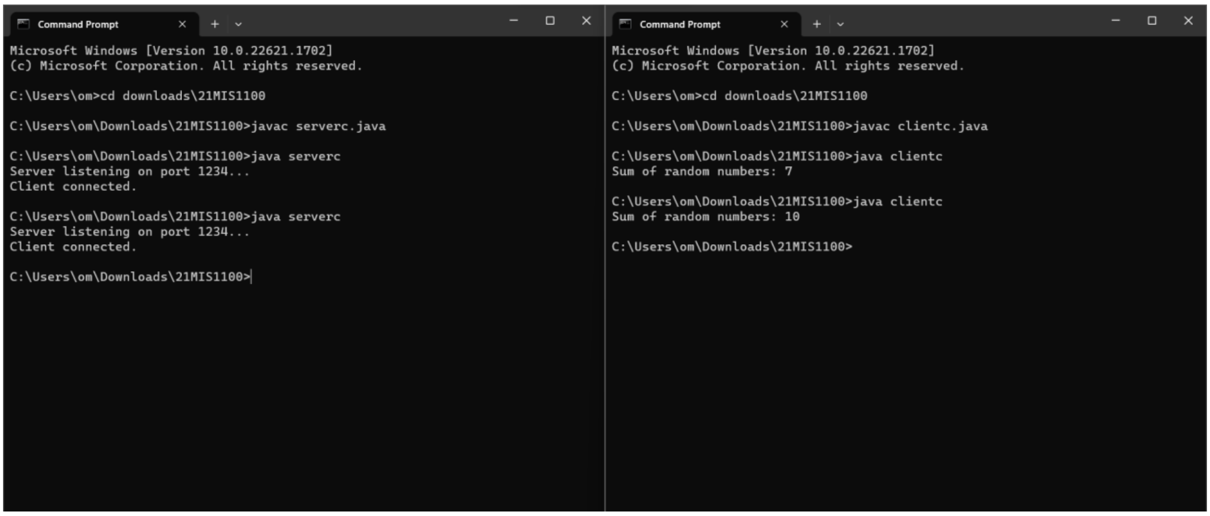
Server:-

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

public class serverc {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234);
            System.out.println("Server listening on port 1234...");

            // Accept client connection
            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected.");

            // Generate random numbers
            int num1 = (int) (Math.random() * 10);
            int num2 = (int) (Math.random() * 10);
            int sum = num1 + num2;

            // Send sum to client
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
true);
            out.println(sum);

            // Close connections
            out.close();
            clientSocket.close();
            serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client:-

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

public class clientc {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234);

            // Receive sum from server
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            int sum = Integer.parseInt(in.readLine());

            // Display sum
            System.out.println("Sum of random numbers: " + sum);

            // Close connection
            in.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:-



```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac serverc.java

C:\Users\om\Downloads\21MIS1100>java serverc
Server listening on port 1234...
Client connected.

C:\Users\om\Downloads\21MIS1100>java serverc
Server listening on port 1234...
Client connected.

C:\Users\om\Downloads\21MIS1100>
```

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac clientc.java

C:\Users\om\Downloads\21MIS1100>java clientc
Sum of random numbers: 7

C:\Users\om\Downloads\21MIS1100>java clientc
Sum of random numbers: 10

C:\Users\om\Downloads\21MIS1100>
```

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

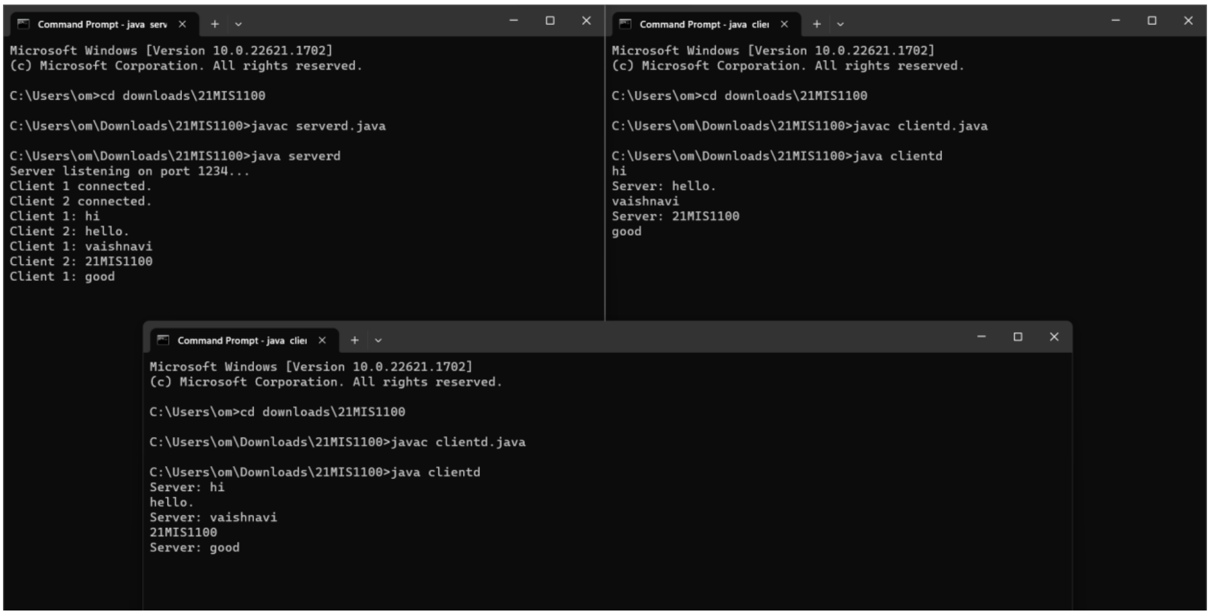
Server:-

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

public class serverd {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234);
            System.out.println("Server listening on port 1234...");

            // Accept client connections
            Socket clientSocket1 = serverSocket.accept();
            System.out.println("Client 1 connected.");
            Socket clientSocket2 = serverSocket.accept();
            System.out.println("Client 2 connected.");
            // Create input/output streams for client 1
            BufferedReader in1 = new BufferedReader(new
InputStreamReader(clientSocket1.getInputStream()));
            PrintWriter out1 = new
PrintWriter(clientSocket1.getOutputStream(), true);

            // Create input/output streams for client 2
            BufferedReader in2 = new BufferedReader(new
InputStreamReader(clientSocket2.getInputStream()));
            PrintWriter out2 = new
PrintWriter(clientSocket2.getOutputStream(), true);

            // Start message exchange
            String client1Message, client2Message;
            while (true) {
                // Receive message from client 1 and send to client 2
                client1Message = in1.readLine();
                if (client1Message != null) {
                    System.out.println("Client 1: " + client1Message);
                    out2.println(client1Message);
                }

                // Receive message from client 2 and send to client 1
                client2Message = in2.readLine();
                if (client2Message != null) {
                    System.out.println("Client 2: " + client2Message);
                    out1.println(client2Message);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client:-

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

public class clientd {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234);

            // Create input/output streams for the server
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            // Start a separate thread to listen for server messages
            Thread serverListener = new Thread(() -> {
                String message;
                try {
                    while ((message = in.readLine()) != null) {
                        System.out.println("Server: " + message);
                    }
                } catch (IOException e) {
                    e.printStackTrace();
                }
            });
            serverListener.start();

            // Read user input and send messages to the server
            Scanner scanner = new Scanner(System.in);
            String input;
            while (true) {
                input = scanner.nextLine();
                out.println(input);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:-



Command Prompt - java serv
Command Prompt - java clie
Command Prompt - java clie

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac serverd.java

C:\Users\om\Downloads\21MIS1100>java serverd
Server listening on port 1234...
Client 1 connected.
Client 2 connected.
Client 1: hi
Client 2: hello.
Client 1: vaishnavi
Client 2: 21MIS1100
Client 1: good
```

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac clientd.java

C:\Users\om\Downloads\21MIS1100>java clientd
hi
Server: hello.
vaishnavi
Server: 21MIS1100
good
```

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac clientd.java

C:\Users\om\Downloads\21MIS1100>java clientd
Server: hi
hello.
Server: vaishnavi
21MIS1100
Server: good
```

(e) Using TCP Sockets write a program to display the current date and time.

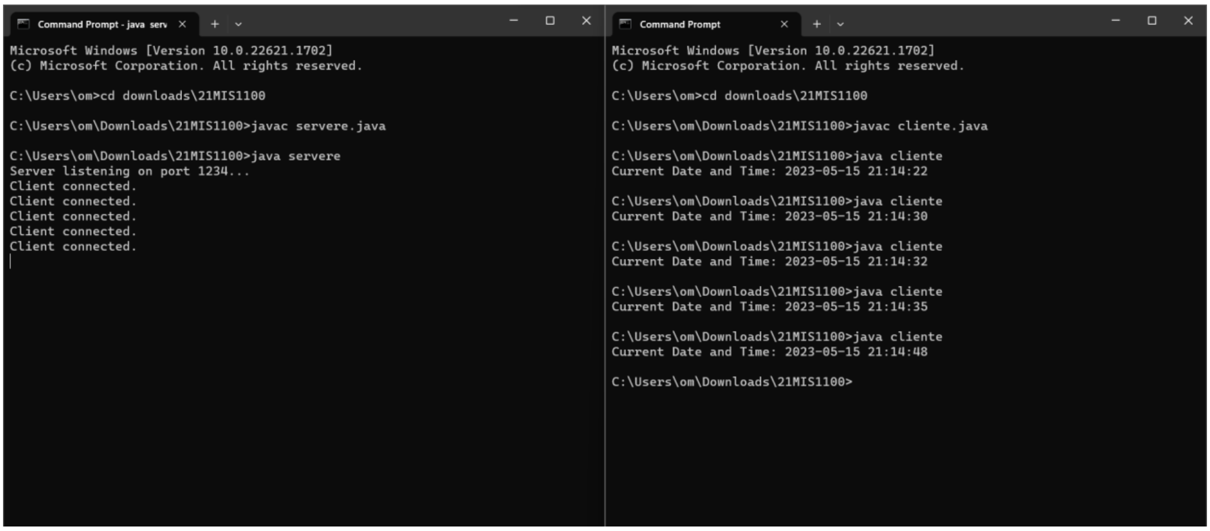
Server:-

```
import java.io.*;
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class servere {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234);
            System.out.println("Server listening on port 1234...");

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Client connected.");

                // Get current date and time
                Date currentDate = new Date();
                SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
                String dateTime = dateFormat.format(currentDate);

                // Send date and time to client
                PrintWriter out = new
PrintWriter(clientSocket.getOutputStream(), true);
                out.println(dateTime);

                // Close connection
                out.close();
                clientSocket.close();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client:-

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

public class cliente {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234);

            // Receive date and time from server
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            String dateTime = in.readLine();

            // Display date and time
            System.out.println("Current Date and Time: " + dateTime);

            // Close connection
            in.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:-



```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac servere.java

C:\Users\om\Downloads\21MIS1100>java servere
Server listening on port 1234...
Client connected.
Client connected.
Client connected.
Client connected.
Client connected.
```

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\om>cd downloads\21MIS1100

C:\Users\om\Downloads\21MIS1100>javac cliente.java

C:\Users\om\Downloads\21MIS1100>java cliente
Current Date and Time: 2023-05-15 21:14:22

C:\Users\om\Downloads\21MIS1100>java cliente
Current Date and Time: 2023-05-15 21:14:30

C:\Users\om\Downloads\21MIS1100>java cliente
Current Date and Time: 2023-05-15 21:14:32

C:\Users\om\Downloads\21MIS1100>java cliente
Current Date and Time: 2023-05-15 21:14:35

C:\Users\om\Downloads\21MIS1100>java cliente
Current Date and Time: 2023-05-15 21:14:48

C:\Users\om\Downloads\21MIS1100>
```