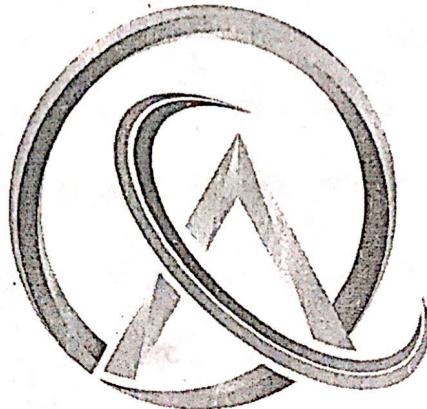


INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT
KALANKI, KATHMANDU
(AFFILIATED TO TRIBHUVAN UNIVERSITY)



ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT

LAB REPORT

SUBJECT : Network Programming
LAB NO : 1

SUBMITTED BY:

NAME : Anish Shrestha
ROLL NO: 1
DATE : 2082/09/30

SUBMITTED TO:

BCA Department

Anish Shrestha

TITLE: INTERNET ADDRESS

OBJECTIVES:

- To understand the InetAddress class and how to resolve hostnames and IP addresses.
- To differentiate between IPv4 and IPv6 addresses.
- To learn to verify domain equality and check the reachability of remote systems.

THEORY:

InetAddress

- The java.net.InetAddress class is used to represent an Internet Protocol (IP) address.
- It does not have public constructors, instead it uses static factory methods like getLocalHost(), getByAddress(), to create instances.

Network Interface

- The java.net.NetworkInterface class represents a network interface consisting of a name and a list of IP addresses assigned to this interface.
- It is used to interrogate the local machine.

IPv4 (Internet Protocol version 4)

- IPv4 is the fourth version of IP that uses 32-bit address scheme allowing for total of 2^{32} addresses.
- Represented in dot-decimal notation (e.g. 192.168.1.1).

IPv6 (Internet Protocol version 6)

- It is designed to solve the address exhaustion of IPv4.
- It uses a 128-bit address scheme, represented as eight groups of four hexadecimal digits (e.g. 2001:0db8:85a3::)

SOURCE CODE:

```
package lab;
import java.io.IOException;
import java.net.*;
import java.util.*;

public class Solution {
    public static void main (String[] args){
        question1();
        question2();
        question3();
        question4();
        question5();
        question6();
    }
}
```

1/2 Program to find hostname from IP address

```
private static void question1() {
    Scanner sc = new Scanner(System.in);
    System.out.println("To find hostname, please enter
                        ip address:");
    String ipAdd = scan.next();
    try {
        InetAddress address = InetAddress.getByName(ipAdd);
        System.out.println("Hostname is:" + address.getHostName());
    } catch (UnknownHostException e) {
        System.out.println("Error:" + e.getMessage());
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent
To find hostname, please enter ip address:
8.8.4.4
Hostname is: dns.google
Process finished with exit code 0
```

1/2 Find IP address of local machine

```
private static void question2() {
    try {
        InetAddress add = InetAddress.getLocalHost();
        System.out.println("IP Address of local machine
                           is:" + add.getHostAddress());
    }
}
```

```
        catch (UnknownHostException e) {
            System.out.println ("Error:" + e.getMessage ());
        }
    }
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-jar
IP Address of local machine is: 192.168.1.4

Process finished with exit code 0
```

113 To verify the given domain are same or not.

```
private static void question3() {
```

```
    String d1 = "google.com";
```

```
    String d2 = "fb.com";
```

```
    try {
```

```
        InetAddress add1 = InetAddress.getByName(d1);
```

```
        InetAddress add2 = InetAddress.getByName(d2);
```

```
        if (add1.getHostName().equals(add2.getHostName()))
            System.out.println (d1 + " and " + d2 + " are same");
```

```
    else
```

```
        System.out.println (d1 + " and " + d2 + " are not
                           same");
```

```
    } catch (UnknownHostException e) {
```

```
        System.out.println ("Error:" + e.getMessage());
```

```
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe
google.com and fb.com are not same!
Process finished with exit code 0"

"C:\Program Files\Java\jdk-23\bin\java.exe
google.com and google.com are same!
Process finished with exit code 0"
```

Q14 To determine IPV4 or IPV6

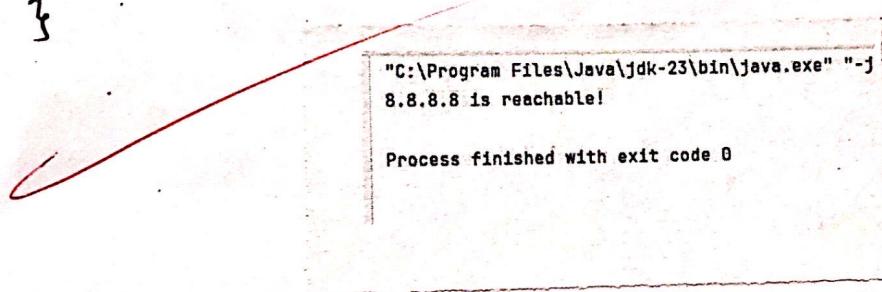
```
private static void question4(){
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter Ip address to check:");
    String ip = sc.nextLine();
    try {
        InetAddress add = InetAddress.getByName (ip);
        if (add instanceof Inet4Address)
            System.out.println (ip + " is IPV4 Address");
        else if (add instanceof Inet6Address)
            System.out.println (ip + " is IPV6 Address");
        else
            System.out.println ("Mixed");
    }
    catch (UnknownHostException e) {
        System.out.println ("Error :" + e.getMessage());
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe
Enter Ip address to check:
192.168.1.1
192.168.1.1 is IPV4 Address!
Process finished with exit code 0"

"C:\Program Files\Java\jdk-23\bin\java.exe
Enter Ip address to check:
2001:4860:4860::8888
2001:4860:4860::8888 is IPV6 Address!
Process finished with exit code 0"
```

115) To check if IP is reachable

```
private static void question5() {
    String ip = "8.8.8.8";
    try {
        InetAddress add = InetAddress.getByName(ip);
        if (add.isReachable(500))
            System.out.println(ip + " is reachable");
        else
            System.out.println(ip + " is not reachable");
    } catch (IOException e) {
        System.out.println("Error: " + e.getMessage());
    }
}
```



116) To list all network interfaces

```
private static void question6() {
    Enumeration<NetworkInterface> nis;
    try {
        nis = NetworkInterface.getNetworkInterfaces();
        while (nis.hasMoreElements()) {
            NetworkInterface n = nis.nextElement();
            System.out.println("Name: " + n.getName());
        }
    }
```

```

        System.out.println("Display name:" + n.getDisplayName());
    }
    System.out.println("Hardware " + n.getHardwareAddress());
}
} catch (SocketException e) {
    System.out.println(e.getMessage());
}
}

```

```

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\downloadssss\IntelliJ IDEA Community Edition :
Name: ethernet_0
Display Name: Killer E2600 Gigabit Ethernet Controller-WFP Native MAC Layer LightWeight Filter-0000
Hardware Address: [B@3feba861
Name: ethernet_1
Display Name: Killer E2600 Gigabit Ethernet Controller-QoS Packet Scheduler-0000
Hardware Address: [B@5b480cf9
Name: ethernet_2
Display Name: Killer E2600 Gigabit Ethernet Controller-WFP 802.3 MAC Layer LightWeight Filter-0000
Hardware Address: [B@6f496d9f
Name: ethernet_3
Display Name: WAN Miniport (IP)-WFP Native MAC Layer LightWeight Filter-0000
Hardware Address: null
Name: ethernet_4
Display Name: WAN Miniport (IP)-QoS Packet Scheduler-0000
Hardware Address: null
Name: ethernet_5
Display Name: WAN Miniport (IPv6)-WFP Native MAC Layer LightWeight Filter-0000
Hardware Address: null
Name: ethernet_6
Display Name: WAN Miniport (IPv6)-QoS Packet Scheduler-0000
Hardware Address: null
Name: ethernet_7
Display Name: WAN Miniport (Network Monitor)-WFP Native MAC Layer LightWeight Filter-0000
Hardware Address: null
Name: ethernet_8
Display Name: WAN Miniport (Network Monitor)-QoS Packet Scheduler-0000

```

CONCLUSION:

In this lab, we successfully implemented Java programs to utilize the `java.net` package. We explored how the `InetAddress` class handles IP versioning. We also learned to check the status of remote hosts using the `isReachable` method.

