

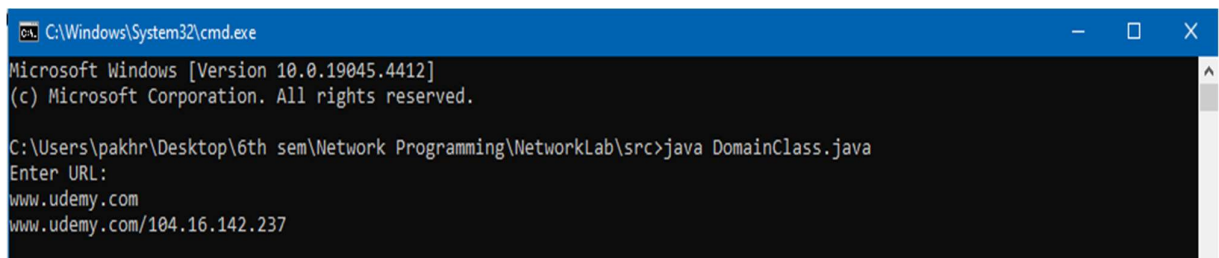
1.1. Write the program that prints the address of www.Udemy.com.

Program:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;

public class DomainClass {
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter URL:");
        String url=sc.next();
        try{
            InetAddress address=InetAddress.getByName(url);
            System.out.println(address);
        }
        catch (UnknownHostException e){
            System.out.println("Invalid "+url);
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java DomainClass.java
Enter URL:
www.udemy.com
www.udemy.com/104.16.142.237
```

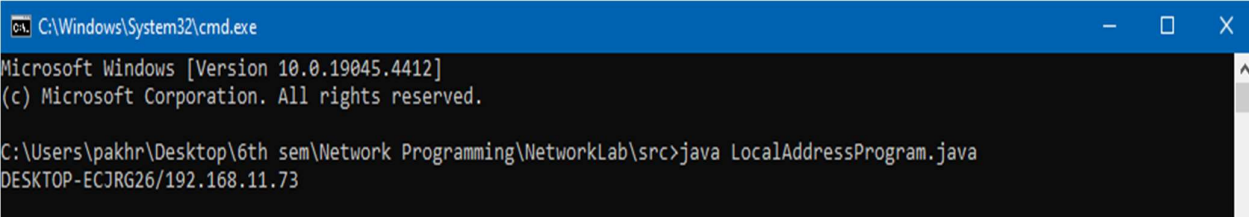
1.2. Write a program to find the address of the local machine.

Program:

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class LocalAddressProgram {
    public static void main(String[] args){
        try{
            InetAddress localIp=InetAddress.getLocalHost();
            System.out.println(localIp);
        }
        catch (UnknownHostException e){
            System.out.println("Man!! I got no address.");
        }
    }
}
```

Output:

A screenshot of a Windows command prompt window. The title bar at the top is blue and contains the text "C:\Windows\System32\cmd.exe" along with standard window control buttons (minimize, maximize, close). The main area of the window is black with white text. It shows the Microsoft Windows version (10.0.19045.4412) and copyright information. The command executed is "java LocalAddressProgram.java" from the directory "C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src". The output of the program is "DESKTOP-ECJRG26/192.168.11.73".

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java LocalAddressProgram.java
DESKTOP-ECJRG26/192.168.11.73
```

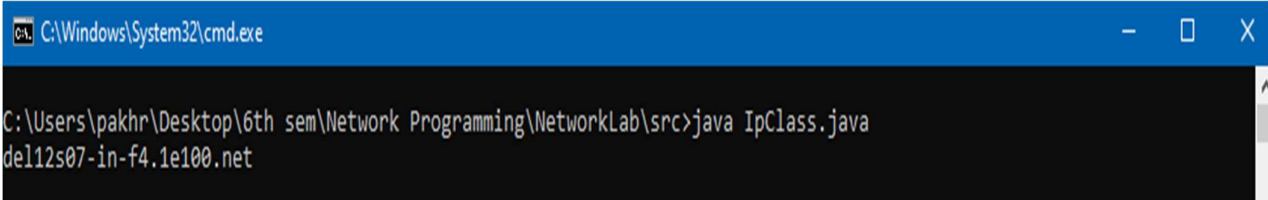
1.3. Write a program to find the hostname with given address.

Program:

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class IpClass {
    public static void main(String[] args){
        try{
            InetAddress
ipAddress=InetAddress.getByName("142.250.194.196");
            System.out.println(ipAddress.getCanonicalHostName());
        }
        catch (UnknownHostException e){
            System.out.println("Invalid IP Address!");
        }
    }
}
```

Output:

A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\Windows\System32\cmd.exe" along with standard window control buttons (minimize, maximize, close). The command prompt area has a black background with white text. The first line shows the command being executed: "C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java IpClass.java". The second line shows the output of the program: "del12s07-in-f4.1e100.net".

```
C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java IpClass.java
del12s07-in-f4.1e100.net
```

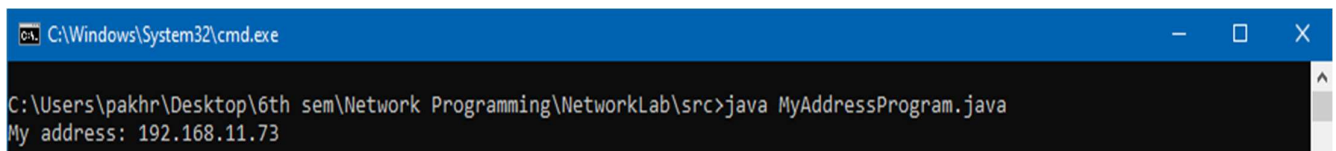
1.4. Write a program to find the IP address of the local machine.

Program:

```
import java.net.*;

public class MyAddressProgram {
    public static void main(String[] args){
        try {
            InetAddress me = InetAddress.getLocalHost();
            String dottedQuad = me.getHostAddress();
            System.out.println("My address: " + dottedQuad);
        } catch (UnknownHostException ex) {
            System.out.println("Myan! I got no address");
        }
    }
}
```

Output:

A screenshot of a Windows command prompt window. The title bar is blue and shows the path 'C:\Windows\System32\cmd.exe'. The command prompt is black with white text. The command entered is 'C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java MyAddressProgram.java'. The output is 'My address: 192.168.11.73'.

```
C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java MyAddressProgram.java
My address: 192.168.11.73
```

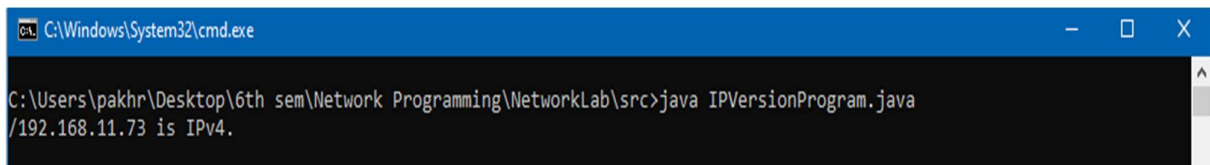
1.5. Write a program to determine whether the address is IPv4 or IPv6.

Program:

```
import java.net.*;

public class IPVersionProgram {
    public static void main(String[] args){
        try {
            InetAddress ia = InetAddress.getByName("192.168.11.73");
            byte[] address = ia.getAddress();
            if(address.length==4)
                System.out.println(ia + " is IPv4.");
            else if(address.length==16)
                System.out.println(ia + " is IPv6.");
        } catch (UnknownHostException ex) {
            System.out.println("Invalid IP!");
        }
    }
}
```

Output:

A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\Windows\System32\cmd.exe" along with standard window control buttons (minimize, maximize, close). The command prompt area is black with white text. The command entered is "C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java IPVersionProgram.java". The output displayed is "192.168.11.73 is IPv4.".

```
C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java IPVersionProgram.java
192.168.11.73 is IPv4.
```

1.6. Write a program to test the characteristics of an IP address

Program:

```
import java.net.*;

public class IPCharacteristics {

    public static void main(String[] args) {

        try {

            InetAddress address =
InetAddress.getByName("www.Udemy.com");

            if (address.isAnyLocalAddress()) {

                System.out.println(address + " is a wildcard address.");

            }

            if (address.isLoopbackAddress()) {

                System.out.println(address + " is a loopback address.");

            }

            if (address.isLinkLocalAddress()) {

                System.out.println(address + " is a link-local address.");

            } else if (address.isSiteLocalAddress()) {

                System.out.println(address + " is a site-local address.");

            } else {

                System.out.println(address + " is a global address.");

            }

            if (address.isMulticastAddress()) {

                if (address.isMCGlobal()) {

                    System.out.println(address + " is a global multicast
address.");

                } else if (address.isMCOrgLocal()) {

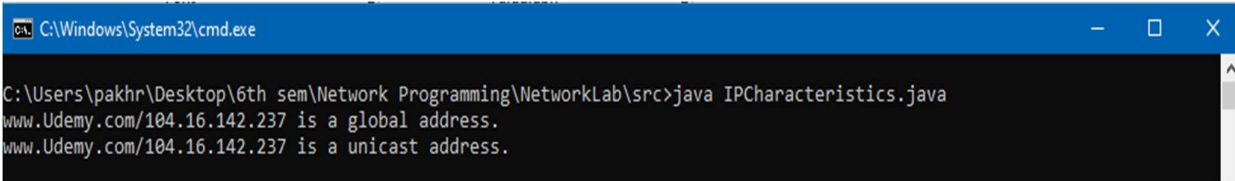
                    System.out.println(address + " is an organization wide
```

```

multicast address.");
        } else if (address.isMCSiteLocal()) {
            System.out.println(address + " is a site wide multicast
address.");
        } else if (address.isMCLinkLocal()) {
            System.out.println(address + " is a subnet wide multicast
address.");
        } else if (address.isMCNodeLocal()) {
            System.out.println(address + " is an interface-local multicast
address.");
        } else {
            System.out.println(address + " is an unknown multicast
address type.");
        }
    } else {
        System.out.println(address + " is a unicast address.");
    }
} catch (UnknownHostException ex) {
    System.err.println("Could not resolve the address.");
}
}
}
}

```

Output:



```

C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java IPCharacteristics.java
www.Udemy.com/104.16.142.237 is a global address.
www.Udemy.com/104.16.142.237 is a unicast address.

```

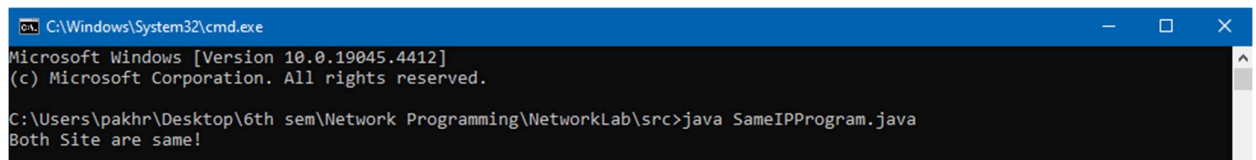
1.7. Write a Program to determine where two different domain name have same IP address.

Program:

```
import java.net.*;

public class SameIPProgram {
    public static void main(String[] args) {
        try {
            InetAddress site1 = InetAddress.getByName("Animension.net");
            InetAddress site2 = InetAddress.getByName("Animension.to");
            if (site1.equals(site2)) {
                System.out.println("Both Site are same!");
            } else {
                System.out.println("Different Site");
            }
        } catch (UnknownHostException ex) {
            System.out.println("Host lookup failed.");
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

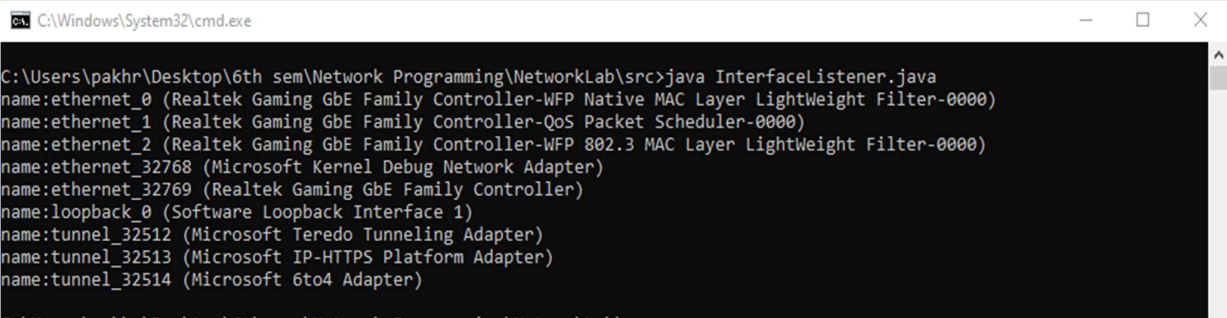
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java SameIPProgram.java
Both Site are same!
```


1.8. Write a program that lists all the network interfaces.

Program:

```
import java.net.*;
import java.util.*;
public class InterfaceListener {
    public static void main(String[] args) throws SocketException {
        Enumeration<NetworkInterface> interfaces =
NetworkInterface.getNetworkInterfaces();
        while (interfaces.hasMoreElements()) {
            NetworkInterface ni = interfaces.nextElement();
            System.out.println(ni);
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java InterfaceListener.java
name:ethernet_0 (Realtek Gaming GbE Family Controller-WFP Native MAC Layer LightWeight Filter-0000)
name:ethernet_1 (Realtek Gaming GbE Family Controller-QoS Packet Scheduler-0000)
name:ethernet_2 (Realtek Gaming GbE Family Controller-WFP 802.3 MAC Layer LightWeight Filter-0000)
name:ethernet_32768 (Microsoft Kernel Debug Network Adapter)
name:ethernet_32769 (Realtek Gaming GbE Family Controller)
name:loopback_0 (Software Loopback Interface 1)
name:tunnel_32512 (Microsoft Teredo Tunneling Adapter)
name:tunnel_32513 (Microsoft IP-HTTPS Platform Adapter)
name:tunnel_32514 (Microsoft 6to4 Adapter)
```

1.9. Write a program to demonstrate SpamCheck.

Program:

```
import java.net.*;

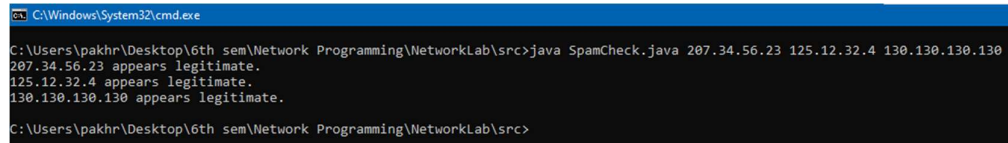
public class SpamCheck {

    public static final String BLACKHOLE = "sbl.spamhaus.org";

    public static void main(String[] args) throws UnknownHostException
    {
        for (String arg: args) {
            if (isSpammer(arg)) {
                System.out.println(arg + " is a known spammer.");
            } else {
                System.out.println(arg + " appears legitimate.");
            }
        }
    }

    private static boolean isSpammer(String arg) {
        try {
            InetAddress address = InetAddress.getByName(arg);
            byte[] quad = address.getAddress();
            String query = BLACKHOLE;
            for (byte octet : quad) {
                int unsignedByte = octet < 0 ? octet + 256 : octet;
                query = unsignedByte + "." + query;
            }
            InetAddress.getByAddress(query);
            return true;
        } catch (UnknownHostException e) {
            return false;
        }
    }
}
```

Output:



```
C:\Windows\System32\cmd.exe
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>java SpamCheck.java 207.34.56.23 125.12.32.4 130.130.130.130
207.34.56.23 appears legitimate.
125.12.32.4 appears legitimate.
130.130.130.130 appears legitimate.
C:\Users\pakhr\Desktop\6th sem\Network Programming\NetworkLab\src>
```