**CSE3011 NETWORK PROGRAMMING**

**LAB EXPERIMENT 4**

NAME – B PRATYUSH

REGISTRATION NUMBER – 19BCN7114

LAB SLOT – L1+L2

FACULTY – PROF. MUNEESWARI

**Experiment Description: IP Characteristics and URL Content Extraction**

**CODE 1**

**URLProg.java**

package Lab4;

import java.net.\*;

import java.util.\*;

import java.io.\*;

public class URLProg {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("URL Info extraction");

System.out.println("How many URL's going to be entered");

Scanner sin=new Scanner(System.in);

int n=sin.nextInt();

for(int i=0;i<n;i++)

{

try

{

System.out.println("Enter URL");

String urlin = sin.next();

URL u =new URL(urlin);

System.out.println("Entered URL is :"+u);

System.out.println("The Scheme/Protocol used is: "+u.getProtocol());

System.out.println("The User info is: "+u.getUserInfo());

String host = u.getHost();

System.out.println("The host is: "+host);

int atSign = host.indexOf('@');

if (atSign != -1)

{

host = host.substring(atSign+1);

System.out.println("The host is " + host);

}

else

{

System.out.println("The host is null.");

}

System.out.println("The port number is " + u.getPort());

System.out.println("The path is " + u.getPath());

System.out.println("The anchor of the URL is " + u.getRef());

System.out.println("The query string is " + u.getQuery());

System.out.println("The Hashcode value is " + u.hashCode());

System.out.println("The URL CONTENT: "+u.getContent());

System.out.println("File Name: "+u.getFile());

}

catch(MalformedURLException e)

{

System.out.println("Cannot understand the entered URL");

System.out.println(e);

}

catch(IOException e)

{

System.out.println(e);

}

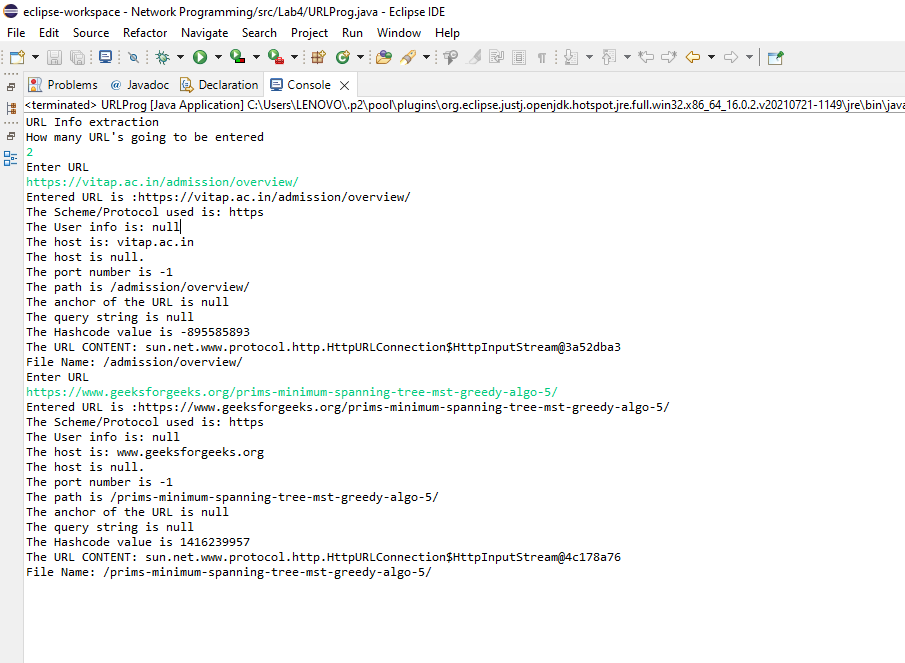
}

sin.close();

}

}

Output



**Code 2**

**IPCharac.java**

package Lab4;

import java.net.InetAddress;

import java.net.\*;

import java.util.Scanner;

import java.io.\*;

public class IPCharac {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Ip characteristics");

System.out.println("Enter number of IP s");

Scanner sin=new Scanner(System.in);

int n=sin.nextInt();

for(int i=0;i<n;i++)

{

try

{

System.out.println("Enter IP");

String ipadd = sin.next();

InetAddress address = InetAddress.getByName(ipadd);

System.out.println("Host Name of IP :"+address.getHostName());

System.out.println("Canonical Host Name of IP :"+address.getCanonicalHostName());

System.out.println("Loopback addres of IP :"+address.getLoopbackAddress());

System.out.println("Is the ip address reachable :"+address.isReachable(300));

if (address.isAnyLocalAddress())

{

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

}

if (address.isLoopbackAddress())

{

System.out.println(address + " is 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 e)

{

System.out.println("Cannot understand the entered IP");

System.out.println(e);

}

catch(IOException e)

{

System.out.println(e);

}

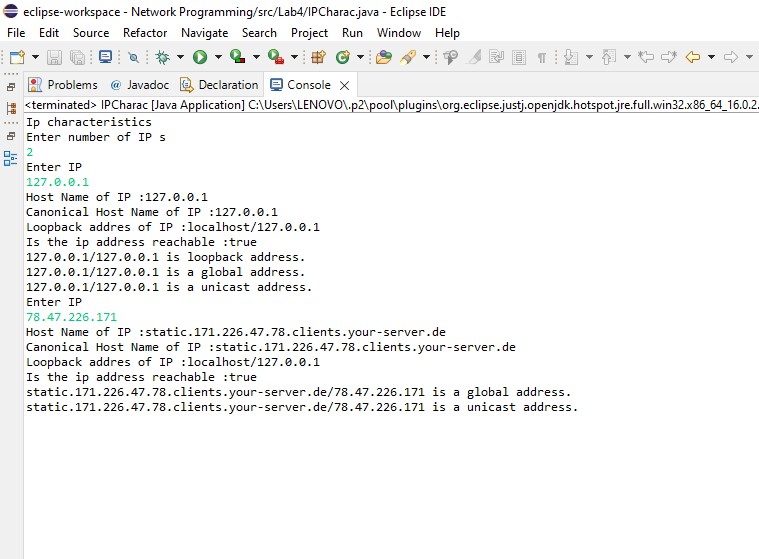
}

sin.close();

}

}

**Output**

****

**Code 3**

**IPUrlProg.java**

package Lab4;

import java.net.\*;

import java.util.\*;

import java.io.\*;

public class IPUrlProg {

public static void main(String args[]) {

System.out.println("URL and IP Info extraction");

System.out.println("Enter number of url addresses");

Scanner sin=new Scanner(System.in);

int n=sin.nextInt();

for(int i=0;i<n;i++)

{

try

{

System.out.println("Enter IP");

String url = sin.next();

URL u = new URL(url);

System.out.println("Entered URL is :"+u);

System.out.println("The Scheme/Protocol used is: "+u.getProtocol());

System.out.println("The User info is: "+u.getUserInfo());

String host = u.getHost();

System.out.println("The host is: "+host);

int atSign = host.indexOf('@');

if (atSign != -1)

{

host = host.substring(atSign+1);

System.out.println("The host is " + host);

}

else

{

System.out.println("The host is null.");

}

System.out.println("The port number is " + u.getPort());

System.out.println("The path is " + u.getPath());

System.out.println("The anchor of the URL is " + u.getRef());

System.out.println("The query string is " + u.getQuery());

System.out.println("The Hashcode value is " + u.hashCode());

System.out.println("The URL CONTENT: "+u.getContent());

System.out.println("File Name: "+u.getFile());

System.out.println();

System.out.println("Passing the url host into getByname method to get IP address to obtain further IP characteristics");

System.out.println();

InetAddress address = InetAddress.getByName(host);

System.out.println("Host Name of IP :"+address.getHostName());

System.out.println("Canonical Host Name of IP :"+address.getCanonicalHostName());

System.out.println("Loopback addres of IP :"+address.getLoopbackAddress());

System.out.println("Is the ip address reachable :"+address.isReachable(300));

if (address.isAnyLocalAddress())

{

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

}

if (address.isLoopbackAddress())

{

System.out.println(address + " is 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(MalformedURLException e)

{

System.out.println("Cannot understand the entered URL");

System.out.println(e);

}

catch(IOException e)

{

System.out.println(e);

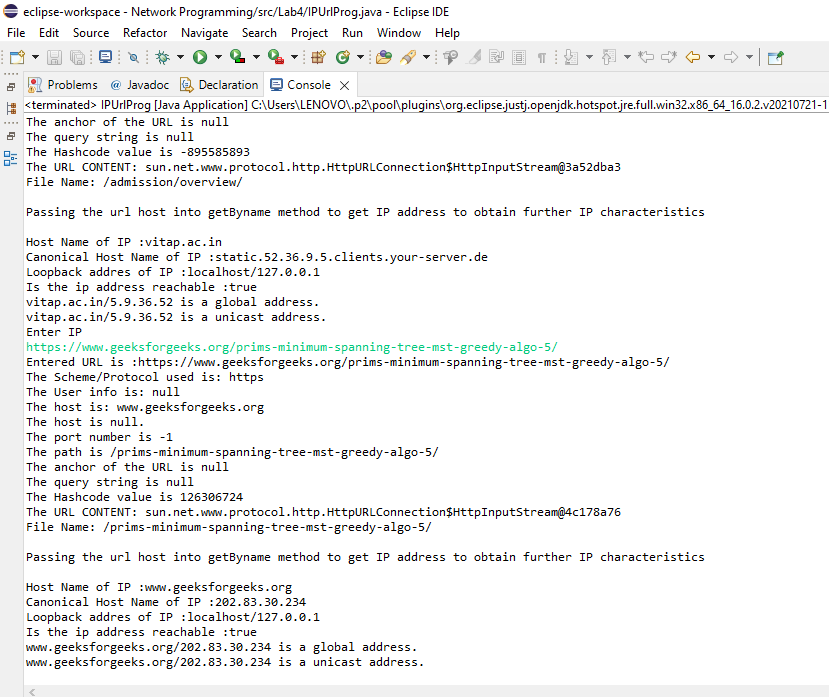
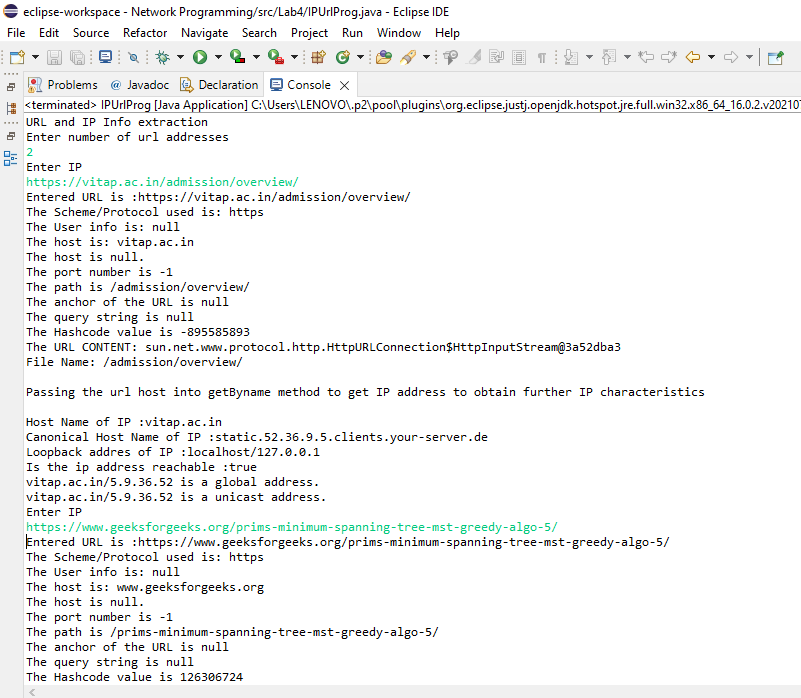
}

}

sin.close();

}

}

**Output**