

Lab No: 1

Title: Write a program that prints the address  
of www.tutohss.edu.in.

Source Code:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class PrintIPAddress
{
    public static void main(String[] args)
    {
        try
        {
            InetAddress address = InetAddress.getByName
                ("www.tutohss.edu.in");
            System.out.println("IP Address: " +
                address.getHostAddress());
        }
        catch (UnknownHostException e)
        {
            System.out.println("Error: Unable to
                resolve hostname");
        }
    }
}
```

Lab No: 4

Title: Write a program to find the IP address and Host name of the local machine

Source Code:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class LocalMachineInfo
{
    public static void main (String [] args)
    {
        try
        {
            InetAddress localAddress = InetAddress.getLocalHost();
            String hostName = localAddress.getHostByName();
            String ipAddress = localAddress.getHostAddress();
            System.out.println ("Local Host Name: " + hostName);
            System.out.println ("Local IP Address: " + ipAddress);
        }
        catch (UnknownHostException e)
        {
            System.out.println ("Error: Unable to determine local IP address and host name");
        }
    }
}
```

Lab No: 4

Title: Write a program to find the IP address and Host name of the local machine

Source Code:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class LocalMachineInfo
{
    public static void main (String [] args)
    {
        try
        {
            InetAddress localAddress = InetAddress.getLocalHost();
            String hostName = localAddress.getHostByName();
            String ipAddress = localAddress.getHostAddress();
            System.out.println ("Local Host Name: " + hostName);
            System.out.println ("Local IP Address: " + ipAddress);
        }
        catch (UnknownHostException e)
        {
            System.out.println ("Error: Unable to determine local IP address and host name");
        }
    }
}
```

Lab No: 5

Title: WAP to get IPv4 and IPv6 address  
of a given web address

Source code:

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

public class Get IP Address {
    public static void main (String [] args) {
        if (args.length > 0) {
            try {
                InetAddress [] address = InetAddress.
                    getAllByName(args[0]);
                System.out.println ("Address for : " + args[0]);
                for (InetAddress address: address) {
                    if (address.getAddress ().length == 4)
                        System.out.println ("IPv4 Address: " + address.
                            getHostAddress ());
                    else if
                        address.getAddress ().length == 16)
                        System.out.println ("IPv6 Address: " + address.
                            getHostAddress ());
                }
            } catch (UnknownHostException e) {
                e.printStackTrace ();
            }
        }
    }
}
```

```
        catch (UnknownHostException e)
    {
        System.out.println("Error: Unable to
                           resolve host " + args[0]);
    }
}

else
{
    System.out.println("Usage : java
                        GetIPAddresses <web address> ");
}
}
}
```

Output: 5

```
PS D:\6sem\NP\assignment> javac GetIPAddresses.java
PS D:\6sem\NP\assignment> java GetIPAddresses
Usage: java GetIPAddresses <webaddress>
PS D:\6sem\NP\assignment> java GetIPAddresses www.google.com
>>
Addresses for: www.google.com
IPv4 Address: 142.250.182.36
PS D:\6sem\NP\assignment> |
```

Lab No: 6

Title: WAP for determining whether an IP address is IPv4 or IPv6

Source Code:

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

public class IPAddressTypeChecker {
    public static void main(String[] args) {
        if (args.length > 0) {
            try {
                InetAddress address = InetAddress.getByName(args[0]);
                if (address.getAddress().length == 4)
                    System.out.println("The IP address"
                        + args[0] + " is IPv4");
                else if (address.getAddress().length
                        == 16)
                    System.out.println("The IP address"
                        + args[0] + " is IPv6");
            } catch (UnknownHostException e) {
            }
        }
    }
}
```

```
else
{ System.out.println("Unknown IP address
    type"); }
}
```

```
catch(UnknownHostException e)
{
    System.err.println("Error: Unable
        to resolve IP address" +
        args[0]); }
}
```

else {

```
System.out.println("Usage: java
    IPAddressTypeChecker <IP Address>");
```

}

}

Output: 6

```
PS D:\6sem\NP\assignment> javac IPAddressTypeChecker.java
PS D:\6sem\NP\assignment> java IPAddressTypeChecker 192.168.1.1
The IP address 192.168.1.1 is IPv4.
PS D:\6sem\NP\assignment>
```

Lab No: 7

Title: WAP for testing the characteristics  
of an IP address.

Source Code:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
public class IPAddressCharacteristic
{
    public static void main(String []
        args)
    {
        if(args.length > 0)
        {
            try
            {
                InetAddress address = InetAddress.
                    getByName(args[0]);
                System.out.println("IP Address : " +
                    address.getHostName());
                if(address.getAddress().length == 4)
                    System.out.println("Type : IPv4");
                else if (address.getAddress().length == 16)
                    System.out.println("Type : IPv6");
                else
                    System.out.println("Type : Unknown");
            }
        }
    }
}
```

```
if (address.isLoopbackAddress ())  
{ System.out.println ("Loop back Address"  
"Yes"); }  
  
else  
{ System.out.println ("Loopback Address: No"); }  
  
if (address.isMulticastAddress ())  
{ System.out.println ("Multicast Address: Yes"); }  
  
else  
{ System.out.println ("Multicast Address: No"); }  
  
if (address.isSiteLocalAddress ())  
{ System.out.println ("Site - Local Address: Yes"); }  
  
else  
{ System.out.println ("Site Local Address: No"); }  
  
catch (UnknownHostException e)  
{ System.out.err.println ("Error: Unable to  
resolve IP address " + args[0]); }  
  
else  
{ System.out.println ("Usage: java InetAddress  
Characteristics <IP Address>"); }  
}
```

Lab No: 8

Title: WAP that compare the domain name  
at "www.google.com" and "stackboke.com"  
the same

Source code:

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

public class DomainNameComparison
{
    public static void main (String [] args)
    {
        if (args.length == 2)
        {
            try
            {
                InetAddress address1 = InetAddress.getByName
                    (args[0]);
                InetAddress address2 = InetAddress.getByName
                    (args[1]);
                if (address1.equals (address2))
                {
                    System.out.println (args[0] + " and " + args[1] + " resolve to the same IP address");
                }
                else
                {
                    System.out.println (args[0] + " and " + args[1] + " resolve to different IP address");
                }
            }
        }
    }
}
```

```
        catch (UnknownHostException e)
    {
        System.out.println ("Error: Unable to resolve
        one or both domain names!");
    }
}
```

else

```
{ System.out.println ("Usage: java
DomainNameComparison <domain1>
<domain2>")
}
}
}
```

Output

```
PS D:\6sem\NP\tmgAnil> javac DomainNameComparison.java
PS D:\6sem\NP\tmgAnil> java DomainNameComparison www.ibiblio.org bellos.ibiblio.org
Error: Unable to resolve one or both domain names.
PS D:\6sem\NP\tmgAnil> java DomainNameComparison www.google.com www.facebook.com
www.google.com and www.facebook.com resolve to different IP addresses.
PS D:\6sem\NP\tmgAnil>
```

Lab No: 9

Title:

Write a program that list all the network interfaces

Source code:

```
import java.net.InetAddress;
import java.net.NetworkInterface;
import java.net.SocketException;
import java.util.Enumeration;

public class ListNetworkInterface
{
    public static void main(String[] args)
    {
        Enumeration<NetworkInterface>
        networkInterfaces = NetworkInterface.
            getNetworkInterfaces();
        if (networkInterfaces != null)
        {
            while (networkInterfaces.hasMore
                Elements())
            {
                NetworkInterface networkInterface =
                networkInterfaces.nextElement();
                System.out.println("Name : " + networkInterface.
                    getName());
                System.out.println("Display Name : "
                    + networkInterface.getDisplayName());
            }
        }
    }
}
```

```
System.out.println ("Is Up:" + networkInterface
    .isUp ());
System.out.println ("Is Loopback:" +  

    networkInterface.isVirtual ());
System.out.println ("Is Point-to-Point:" +  

    networkInterface.isPointToPoint ());
System.out.println ("IP Addresses:");
Enumeration<InetAddress>
addresses = networkInterface.getInetAddresses ();
while (addresses.hasMoreElements ()) {
    InetAddress address = addresses.nextElement ();
    System.out.println (address.getHostAddress ());
}
System.out.println ();
else
{
    System.out.println ("No network interfaces");
}
catch (SocketException e)
{
    System.out.println ("Error while trying to  

    retrieve network interfaces:");
    e.printStackTrace ();
}
```

Lab No: 10

Title: WAP a program that splits the parts of a URL [splitting URL into pieces information].

Source Code:

```
import java.net.URL;
import java.net.MalformedURLException;
public class URLSplitter
{
    public static void main(String[] args)
    {
        if(args.length > 0)
        {
            URL url = new URL(args[0]);
            System.out.println("URL: " + url.toString());
            System.out.println("Protocol: " + url.getProtocol());
            System.out.println("Host: " + url.getHost());
            System.out.println("Port: " + url.getPort() == -1 ? "Default": url.getPort());
            System.out.println("Path: " + url.getPath());
            System.out.println("Query: " + url.getQuery());
            System.out.println("Fragment: " + url.getRef());
        }
        catch(MalformedURLException e)
        {
            System.out.println("Error: Invalid URL format");
        }
        System.out.println("Usage: java URLSplitter
<URL>");
```

Lab No: 11

Title: WAP that checks the which protocols does a virtual machine support or not

Source Code:

```
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.Socket;
import java.net.URL;

public class ProtocolSupportChecker
{
    public static void main(String[] args)
    {
        if (args.length != 1)
        {
            System.out.println("Usage :java Protocol
Support Checker <host:port>");
            return;
        }
        String[] addressParts = args[0].split(":");
        if (addressParts.length != 2)
        {
            System.out.println("Error: Please provide
address in the format host:port");
            return;
        }
        String host = addressParts[0];
        int port;
        try
        {
            port = Integer.parseInt(addressParts[1]);
        }
        catch (NumberFormatException e)
        {
            System.out.println("Error: Invalid port
number");
            return;
        }
        System.out.println("Error: Invalid port checking
supported protocols for: " + host + ":" + port);
    }
}
```

```
checkProtocol ("HTTP", "http://" + host + ":" + port);
checkProtocol ("HTTPS", "https://" + host + ":" + port);
checkProtocol ("FTP", "ftp://" + host + ":" + port);}

private static void checkProtocol (String protocolName,
String urlString)
{
try
{
URL url = new URL(urlString);
if (protocolName.equals ("HTTP") ||
protocolName.equals ("HTTPS"))
{
HttpURLConnection connection = (HttpURLConnection)
url.openConnection ();
connection.setRequestMethod ("GET");
connection.setConnectTimeout (3000);
connection.connect ();
int responseCode = connection.getResponseCode ();
if (responseCode == HttpURLConnection.HTTP_OK)
System.out.println (protocolName + " supported");
else
System.out.println (protocolName + " not supported.
Response code : " + responseCode);
connection.disconnect ();
}
else
{
try (Socket socket = new Socket (url.getHost (),
url.getPort ()))
{
System.out.println (protocolName + " supported");
}
catch (IOException e)
{
System.out.println (protocolName + " not supported");
}
}
}
catch (IOException e)
{
System.out.println (protocolName + " not supported");
}
}
```

Lab No: 12

Title: WAP to download a web page of a given web address..

Source Code:

```
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class WebPageDownloader
{
    public static void main(String[] args)
    {
        if(args.length != 2)
        {
            System.out.println("Usage :java WebPageDownloader
<URL> <output file>");
            return;
        }

        String urlString = args[0];
        String outputFile = args[1];
        try
        {
            URL url = new URL(urlString);
            HttpURLConnection connection = (HttpURLConnection)url.openConnection();
            connection.setRequestMethod("GET");
            int responseCode = connection.getResponseCode();
            if(responseCode == HttpURLConnection.HTTP_OK)
            {
                BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
                FileWriter fileWriter = new FileWriter(outputFile);
                String inputLine;
                while((inputLine = in.readLine()) != null)
                {
                    fileWriter.write(inputLine);
                    fileWriter.newLine();
                }
                in.close();
                fileWriter.close();
            }
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
String inputLine;
while ((inputLine = in.readLine()) != null)
    fileWriter.write(inputLine + "\n");
}

in.close();
fileWriter.close();
System.out.println("Web page downloaded successfully
to " + outputFile);
else {
    System.err.println("Failed to download web page.
HTTP response code :" + responseCode);
}

catch (IOException e) {
    System.err.println("Error occurred :" +
e.getMessage());
    e.printStackTrace();
}
```

Lab No: 13  
Title: WAP for resolving relative URI

Source Code:

```
import java.net.URI;
import java.net.URISyntaxException;

public class ResolveRelativeURI
{
    public static void main (String [] args)
    {
        if (args.length != 2)
            System.out.println ("Usage: java ResolveRelative
URI <base-URI> <relative-URI>");

        return;
    }

    URI baseUri = new URI (args[0]);
    URI relativeUri = new URI (args[1]);
    URI resolvedUri = baseUri.resolve (relativeUri);
    System.out.println ("Base URI: " + baseUri);
    System.out.println ("Relative URI: " + relativeUri);
    System.out.println ("Resolved URI: " + resolvedUri);

    catch (URISyntaxException e)
    {
        System.out.println ("Error: Invalid URI
syntax");
        e.printStackTrace ();
    }
}
```

Lab No: 14

Title: WAP to download an object

Source code:

```
import java.net.ssl.*;  
import java.io.BufferedInputStream;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.security.cert.X509Certificate;  
import java.net.HttpURLConnection;  
import java.net.URL;
```

```
public class ObjectDownloader
```

```
{ public static void main (String[] args)
```

```
{ if (args.length != 2)
```

```
{ System.out.println ("Usage : java Object  
Downloader <URL> <output_file>");
```

```
return; }
```

```
String urlString = args[0];
```

```
String outputFile = args[1];
```

```
try { TrustManager[] trustAll = new TrustManager[]
```

```
{ new X509TrustManager []  
getAcceptedIssuers ()
```

```
{ return null; }
```

```
public void checkClientTrusted (X509  
certified cert, String authType)
```

```
{
```

```
}
```

```

public void checkServerTrusted(X509Certificate[] certs, String authType) {
}

SSLContext sc = SSLContext.getInstance("SSL");
sc.init(null, trustAll, new java.security.SecureRandom());
HttpURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
HttpURLConnection.setDefaultHostnameVerifier(new HostnameVerifier() {
    public boolean verify(String hostname,
                          SSLSocket session) {
        return true;
    }
});

URL url = new URL(urlString);
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.setRequestMethod("GET");
int responseCode = connection.getResponseCode();
HTTP_01c)

try (InputStream inputStream = new
      BufferedInputStream(connection.getInputStream)) {
}

```

```
FileOutputStream fileOutputStream = new  
FileOutputStream(outputFile)  
{ byte[] buffer = new byte[1024];  
int bytesRead;  
while (bytesRead = inputStream.read  
(buffer) != -1)  
{ fileOutputStream.write(buffer, 0, bytesRead);  
}  
System.out.println("Object downloaded  
successfully to " + outputFile);  
  
} else {  
System.out.println("Failed to download object.  
HTTP response code: " + responseCode);  
  
} catch (IOException e)  
{ System.out.println("Error occurred: " +  
e.getMessage());  
e.printStackTrace(); }  
catch (Exception e)  
{ System.out.println("Error occurred: " +  
e.getMessage());  
e.printStackTrace(); }  
}
```

Lab No: 15

Title: WAP to demonstrate the de-www-form-urlencoded strings.

Source Code:

```
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;

public class XwwwFormUrlencodedDemo
{
    public static void main(String[] args)
    {
        String urlString = "https://httpbin.org/post";
        String postData = "";
        try
        {
            postData = "name=" + URLEncoder.encode("John Doe", StandardCharsets.UTF_8.to
                String()) + "&age=" + URLEncoder.encode("30", StandardCharsets.UTF_8.to
                String()) + "&city=" + URLEncoder.encode("New York",
                StandardCharsets.UTF_8.to
                String());
            catch (UnsupportedEncodingException e)
            {
                e.printStackTrace();
            }
        }
        try
        {
            URL url = new URL(urlString);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
        }
    }
}
```

Lab No: 16

Title: WAP that communicating with server  
Side Program through GET

Source Code:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URL;

public class GetRequestDemo
{
    public static void main (String [] args)
    {
        String urlString = "https://jsonplaceholder.
typerode.com/posts/2";
        try {
            URL url=new URL(urlString);
            HttpURLConnection connection=(HttpURLConnection)
            url.openConnection();
            connection.setRequestMethod ("GET");
            int responseCode=connection.getResponseCode();
            System.out.println ("Response Code:" + responseCode);
            BufferedReader br=new BufferedReader(
            new InputStreamReader(connection.getInputStream()));
            String responseLine;
            while ((responseLine=br.readLine ())!=null)
            {
                response.append (responseLine.trim ());
            }
        }
    }
}
```

```
        system.out.println ("Response : " + response.toString ());
    }
}

catch (IOException e)
{
    e.printStackTrace ();
}
}
```

Output: 16

```
PS D:\6sem\WP\tmgAnil> javac GetRequestDemo.java
PS D:\6sem\WP\tmgAnil> java GetRequestDemo
Response Code: 200
Response: {"userId": 1, "id": 1, "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"}
PS D:\6sem\WP\tmgAnil>
```