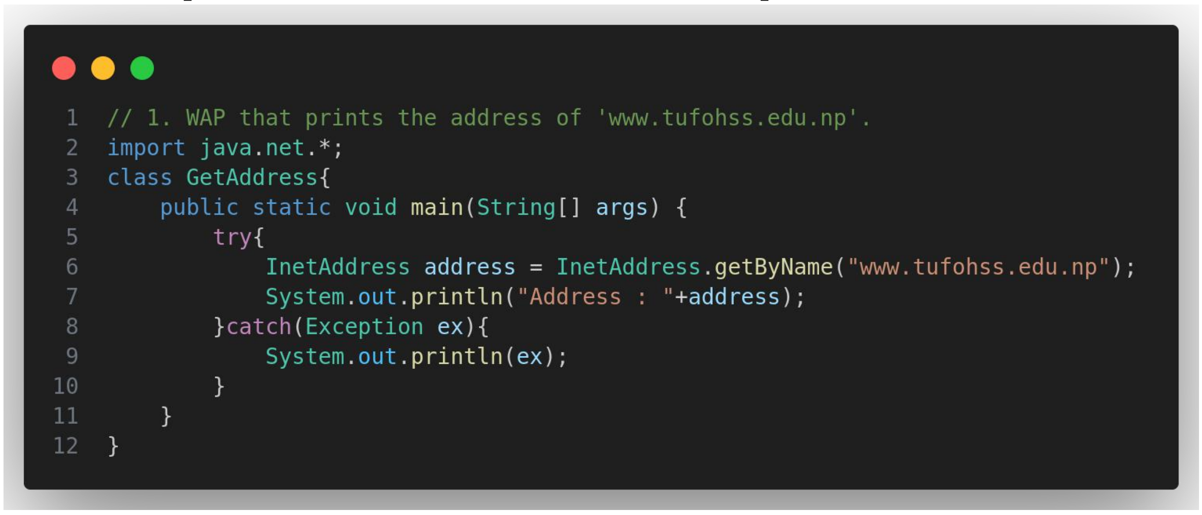
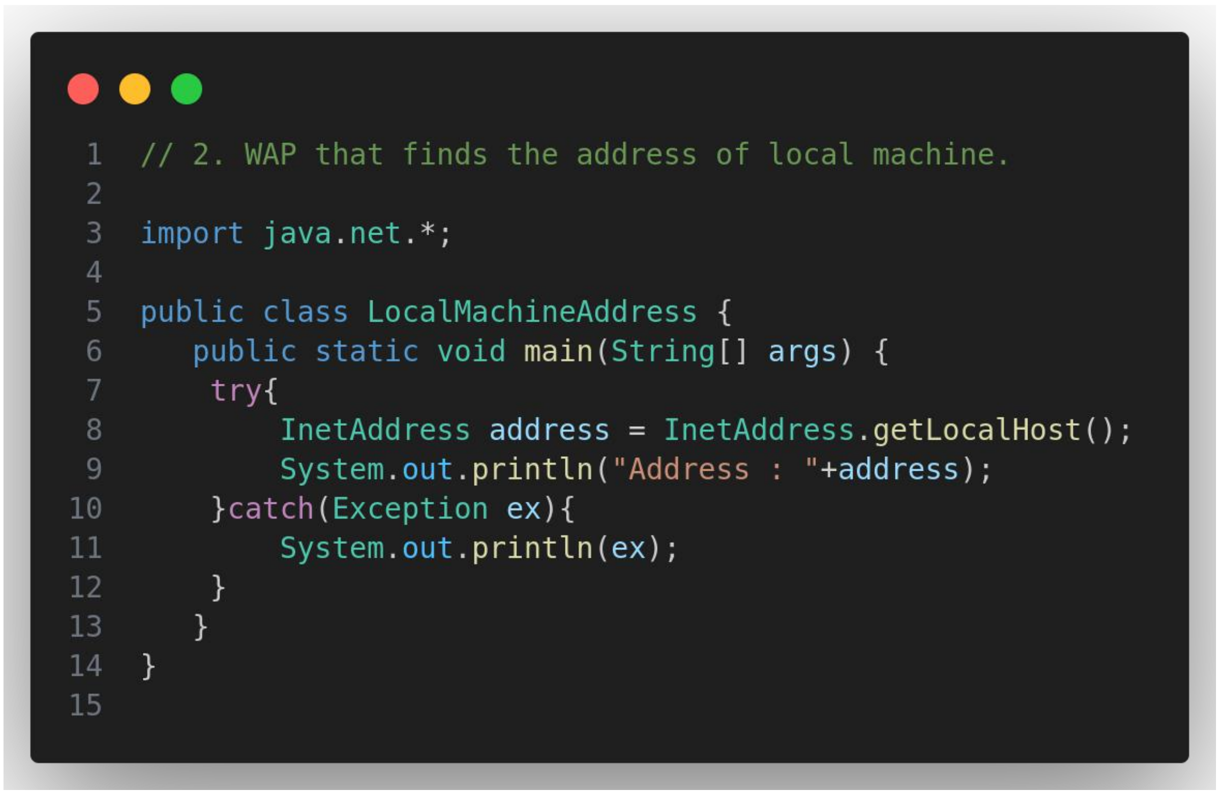
Unit - 2

1. WAP that prints the address of 'www.tufohss.edu.np'.



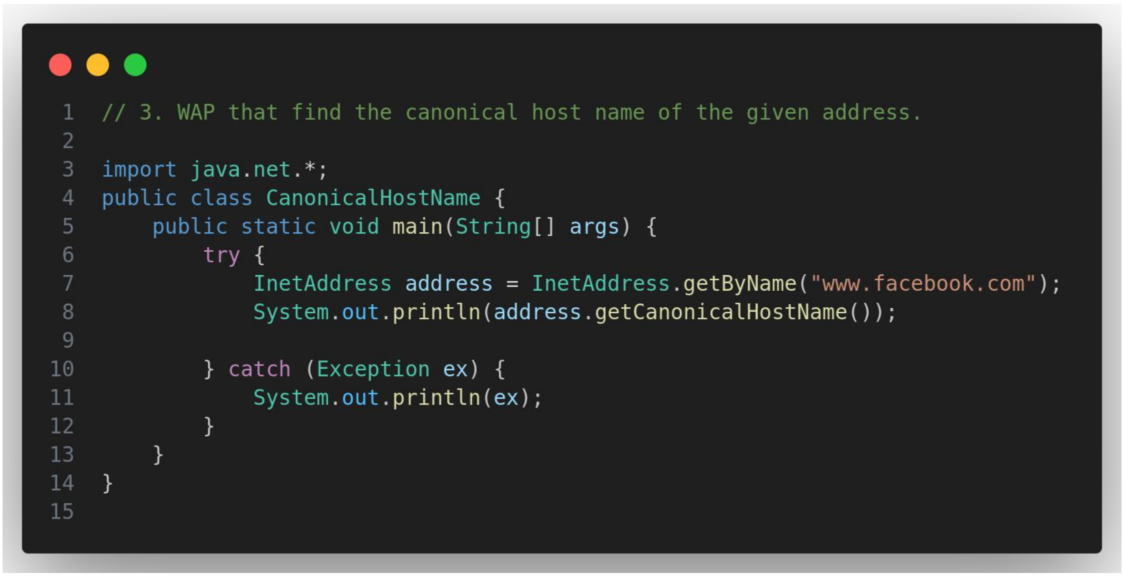
```java
// 1. WAP that prints the address of 'www.tufohss.edu.np'.
import java.net.*;
class GetAddress{
    public static void main(String[] args) {
        try{
            InetAddress address = InetAddress.getByName("www.tufohss.edu.np");
            System.out.println("Address : "+address);
        }catch(Exception ex){
            System.out.println(ex);
        }
    }
}
```

2. WAP that finds the address of local machine.



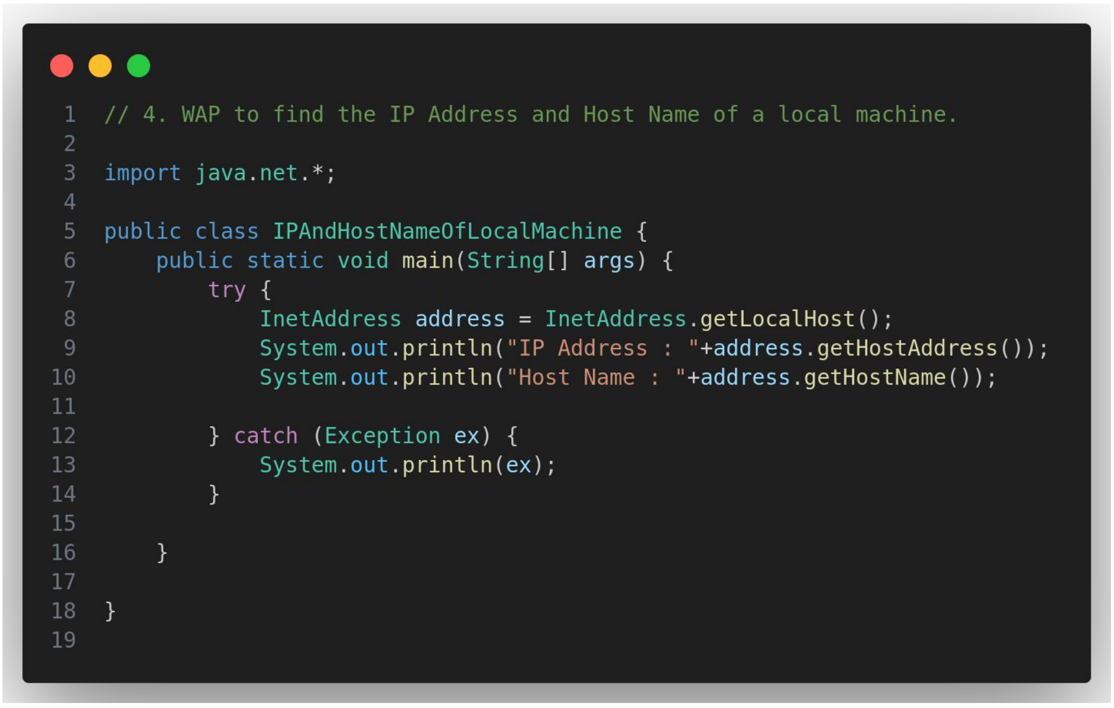
```java
// 2. WAP that finds the address of local machine.

import java.net.*;

public class LocalMachineAddress {
   public static void main(String[] args) {
    try{
        InetAddress address = InetAddress.getLocalHost();
        System.out.println("Address : "+address);
    }catch(Exception ex){
        System.out.println(ex);
    }
   }
}

```

3. WAP that find the canonical host name of the given address.



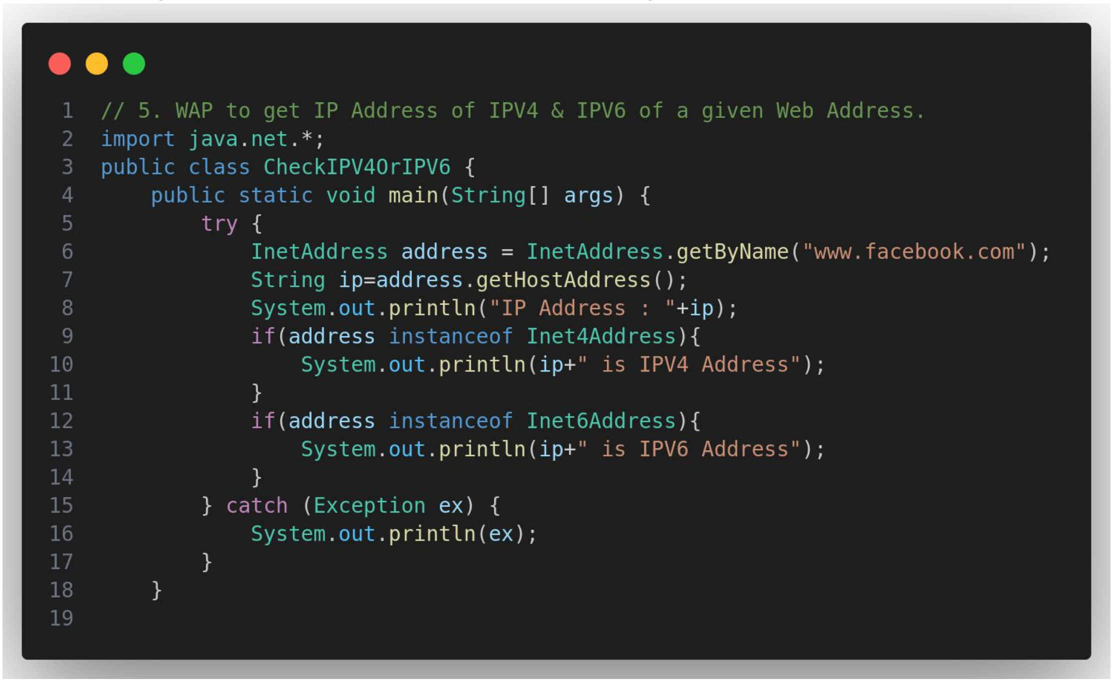
```java
// 3. WAP that find the canonical host name of the given address.

import java.net.*;
public class CanonicalHostName {
    public static void main(String[] args) {
        try {
            InetAddress address = InetAddress.getByName("www.facebook.com");
            System.out.println(address.getCanonicalHostName());

        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

4. WAP to find the IP Address and Host Name of a local machine.



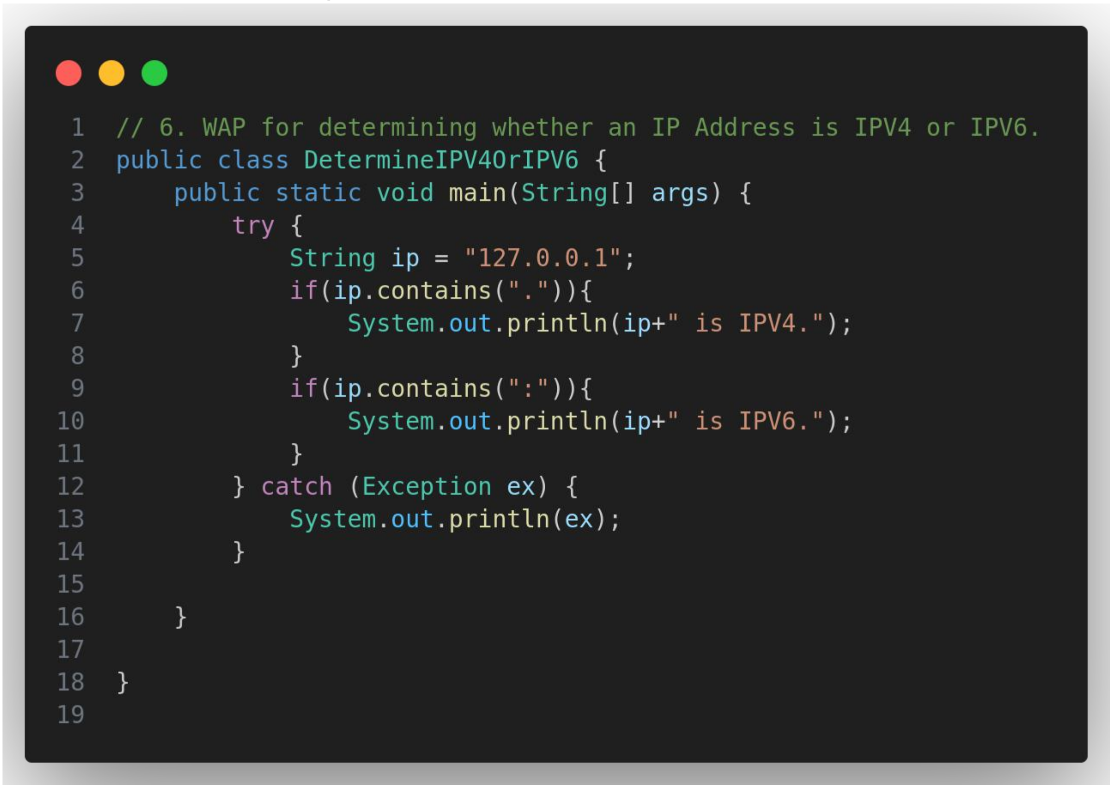
```java
// 4. WAP to find the IP Address and Host Name of a local machine.

import java.net.*;

public class IPAndHostNameOfLocalMachine {
    public static void main(String[] args) {
        try {
            InetAddress address = InetAddress.getLocalHost();
            System.out.println("IP Address : "+address.getHostAddress());
            System.out.println("Host Name : "+address.getHostName());

        } catch (Exception ex) {
            System.out.println(ex);
        }

    }

}

```

5. WAP to get IP Address of IPV4 & IPV6 of a given Web Address.



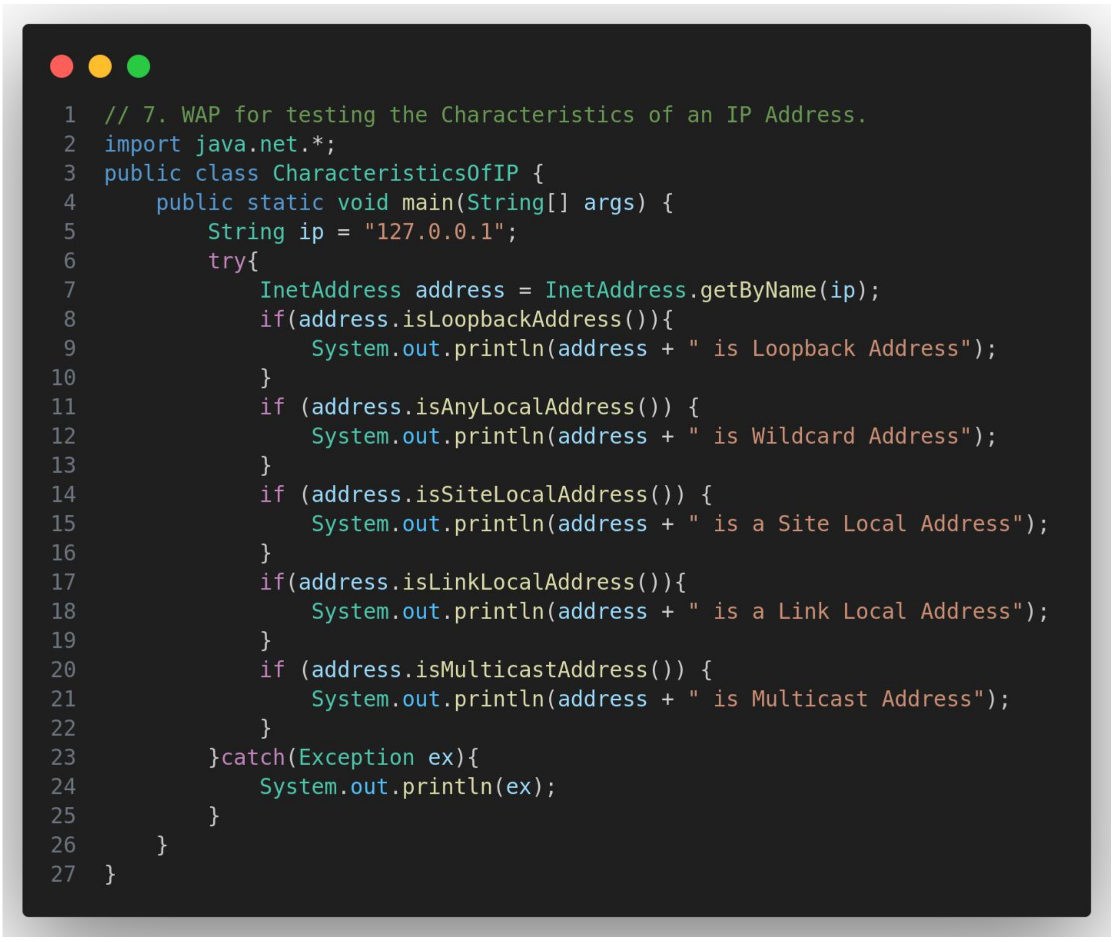
```java
// 5. WAP to get IP Address of IPV4 & IPV6 of a given Web Address.
import java.net.*;
public class CheckIPV4OrIPV6 {
    public static void main(String[] args) {
        try {
            InetAddress address = InetAddress.getByName("www.facebook.com");
            String ip=address.getHostAddress();
            System.out.println("IP Address : "+ip);
            if(address instanceof Inet4Address){
                System.out.println(ip+" is IPV4 Address");
            }
            if(address instanceof Inet6Address){
                System.out.println(ip+" is IPV6 Address");
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

```

6. WAP for determining whether an IP Address is IPV4 or IPV6.



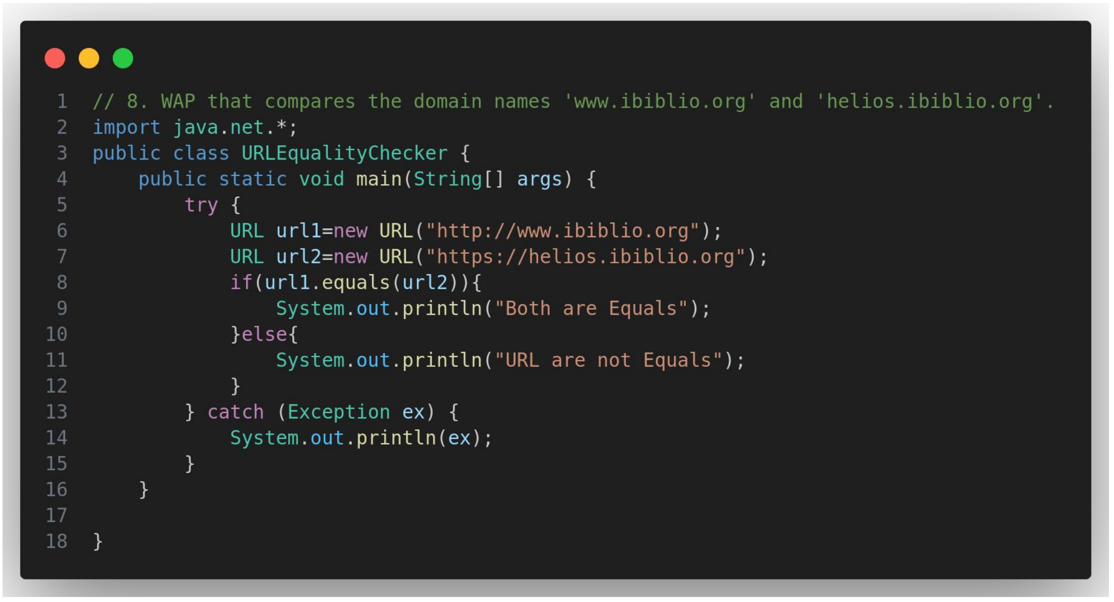
```java
// 6. WAP for determining whether an IP Address is IPV4 or IPV6.
public class DetermineIPV4OrIPV6 {
    public static void main(String[] args) {
        try {
            String ip = "127.0.0.1";
            if(ip.contains(".")){
                System.out.println(ip+" is IPV4.");
            }
            if(ip.contains(":")){
                System.out.println(ip+" is IPV6.");
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }

    }

}

```

7. WAP for testing the Characteristics of an IP Address.



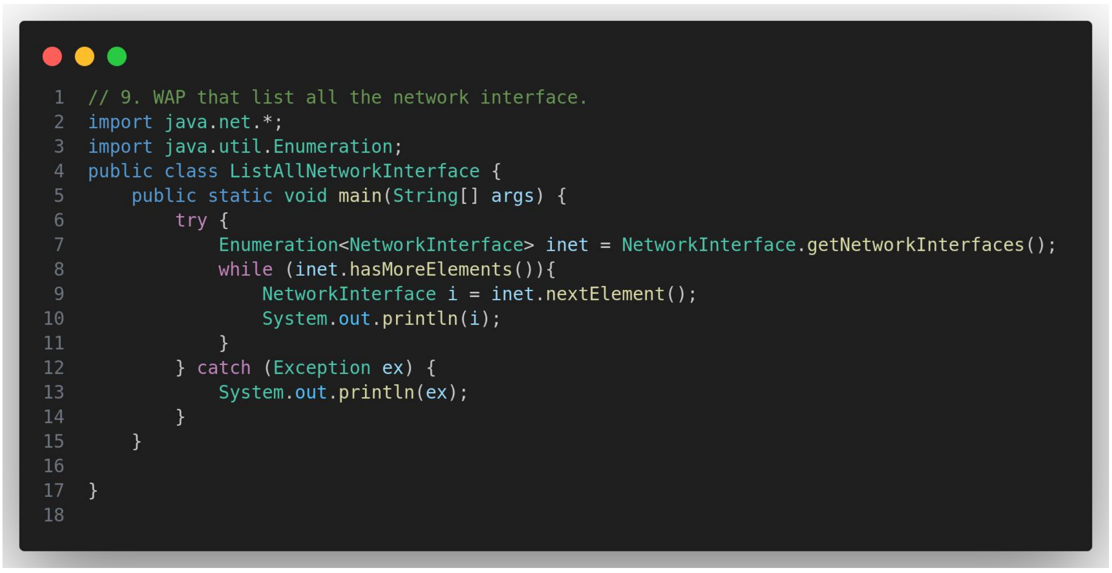
```java
// 7. WAP for testing the Characteristics of an IP Address.
import java.net.*;
public class CharacteristicsOfIP {
    public static void main(String[] args) {
        String ip = "127.0.0.1";
        try{
            InetAddress address = InetAddress.getByName(ip);
            if(address.isLoopbackAddress()){
                System.out.println(address + " is Loopback Address");
            }
            if (address.isAnyLocalAddress()) {
                System.out.println(address + " is Wildcard Address");
            }
            if (address.isSiteLocalAddress()) {
                System.out.println(address + " is a Site Local Address");
            }
            if(address.isLinkLocalAddress()){
                System.out.println(address + " is a Link Local Address");
            }
            if (address.isMulticastAddress()) {
                System.out.println(address + " is Multicast Address");
            }
        }catch(Exception ex){
            System.out.println(ex);
        }
    }
}
```

8. WAP that compares the domain names 'www.ibiblio.org' and 'helios.ibiblio.org'.



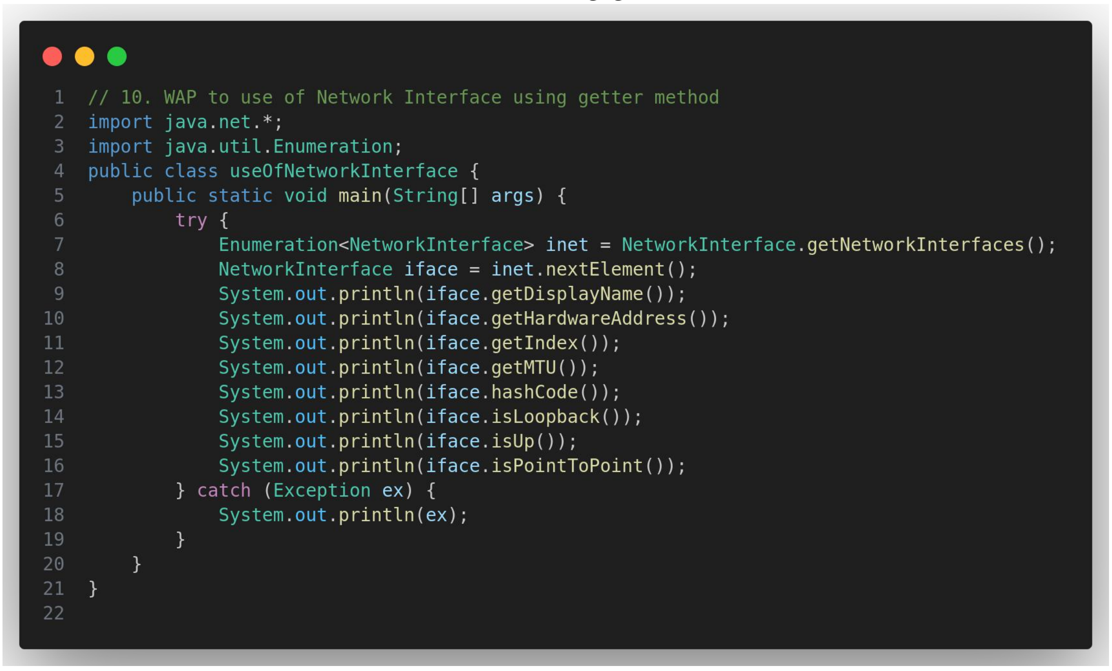
```java
// 8. WAP that compares the domain names 'www.ibiblio.org' and 'helios.ibiblio.org'.
import java.net.*;
public class URLEqualityChecker {
    public static void main(String[] args) {
        try {
            URL url1=new URL("http://www.ibiblio.org");
            URL url2=new URL("https://helios.ibiblio.org");
            if(url1.equals(url2)){
                System.out.println("Both are Equals");
            }else{
                System.out.println("URL are not Equals");
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

}
```

9. WAP that list all the network interface.



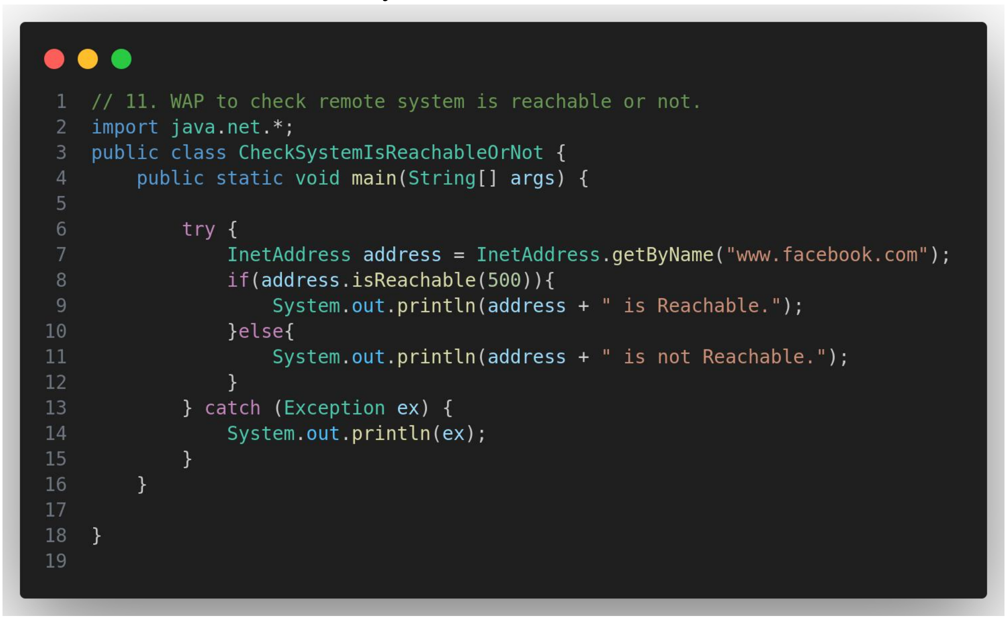
```java
// 9. WAP that list all the network interface.
import java.net.*;
import java.util.Enumeration;
public class ListAllNetworkInterface {
    public static void main(String[] args) {
        try {
            Enumeration<NetworkInterface> inet = NetworkInterface.getNetworkInterfaces();
            while (inet.hasMoreElements()){
                NetworkInterface i = inet.nextElement();
                System.out.println(i);
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

}

```

10. WAP to use of Network Interface using getter method.



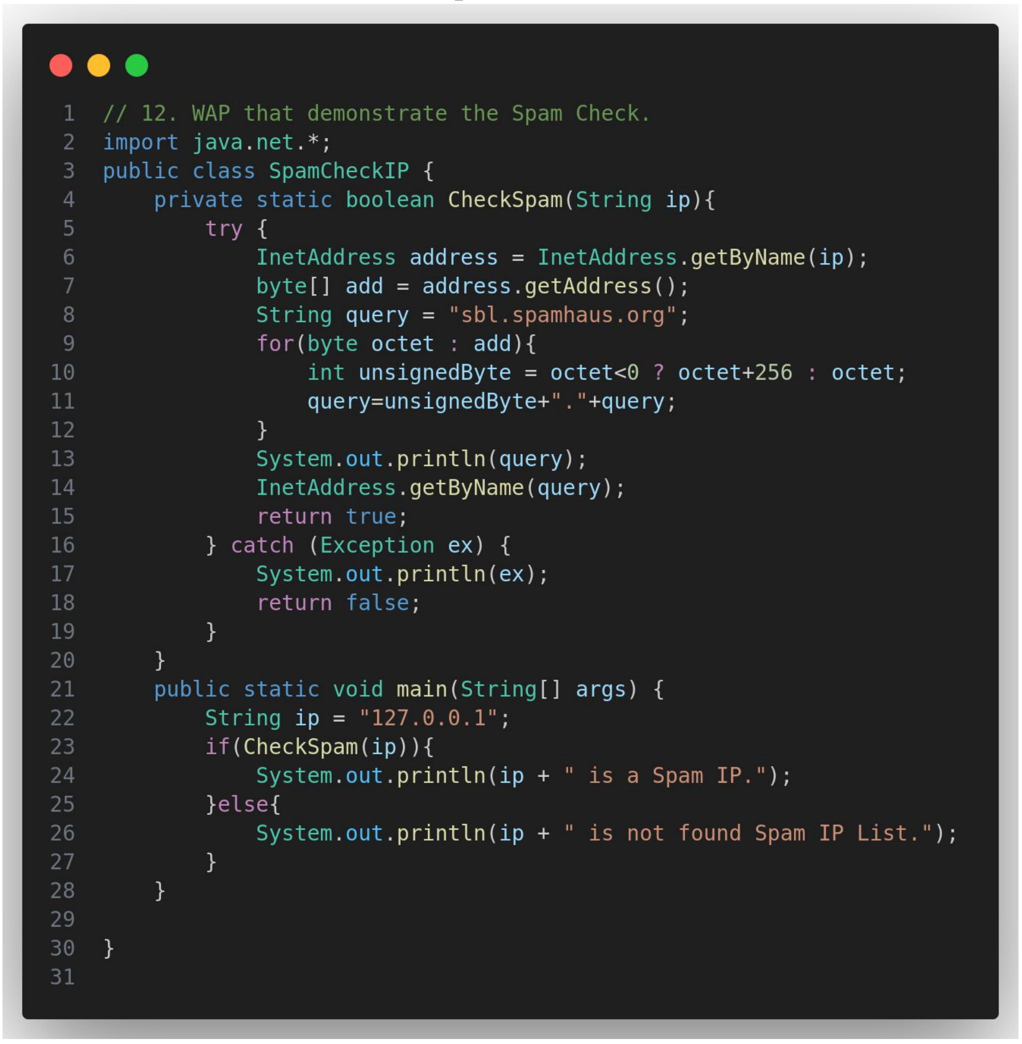
```java
// 10. WAP to use of Network Interface using getter method
import java.net.*;
import java.util.Enumeration;
public class useOfNetworkInterface {
    public static void main(String[] args) {
        try {
            Enumeration<NetworkInterface> inet = NetworkInterface.getNetworkInterfaces();
            NetworkInterface iface = inet.nextElement();
            System.out.println(iface.getDisplayName());
            System.out.println(iface.getHardwareAddress());
            System.out.println(iface.getIndex());
            System.out.println(iface.getMTU());
            System.out.println(iface.hashCode());
            System.out.println(iface.isLoopback());
            System.out.println(iface.isUp());
            System.out.println(iface.isPointToPoint());
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

11. WAP to check remote system is reachable or not.



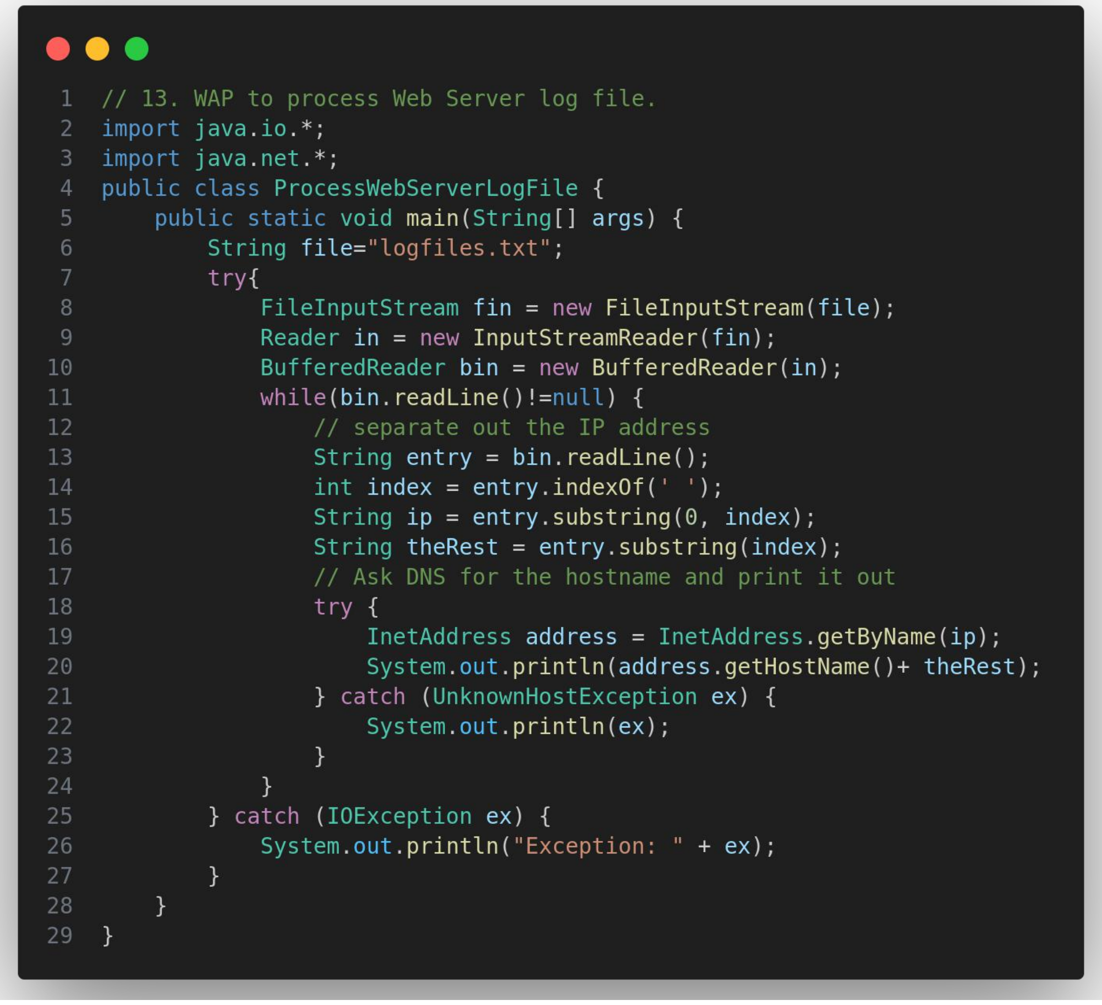
```java
// 11. WAP to check remote system is reachable or not.
import java.net.*;
public class CheckSystemIsReachableOrNot {
    public static void main(String[] args) {

        try {
            InetAddress address = InetAddress.getByName("www.facebook.com");
            if(address.isReachable(500)){
                System.out.println(address + " is Reachable.");
            }else{
                System.out.println(address + " is not Reachable.");
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

}

```

12. WAP that demonstrate the Spam Check.



```java
// 12. WAP that demonstrate the Spam Check.
import java.net.*;
public class SpamCheckIP {
    private static boolean CheckSpam(String ip){
        try {
            InetAddress address = InetAddress.getByName(ip);
            byte[] add = address.getAddress();
            String query = "sbl.spamhaus.org";
            for(byte octet : add){
                int unsignedByte = octet<0 ? octet+256 : octet;
                query=unsignedByte+"."+query;
            }
            System.out.println(query);
            InetAddress.getByName(query);
            return true;
        } catch (Exception ex) {
            System.out.println(ex);
            return false;
        }
    }
    public static void main(String[] args) {
        String ip = "127.0.0.1";
        if(CheckSpam(ip)){
            System.out.println(ip + " is a Spam IP.");
        }else{
            System.out.println(ip + " is not found Spam IP List.");
        }
    }

}

```

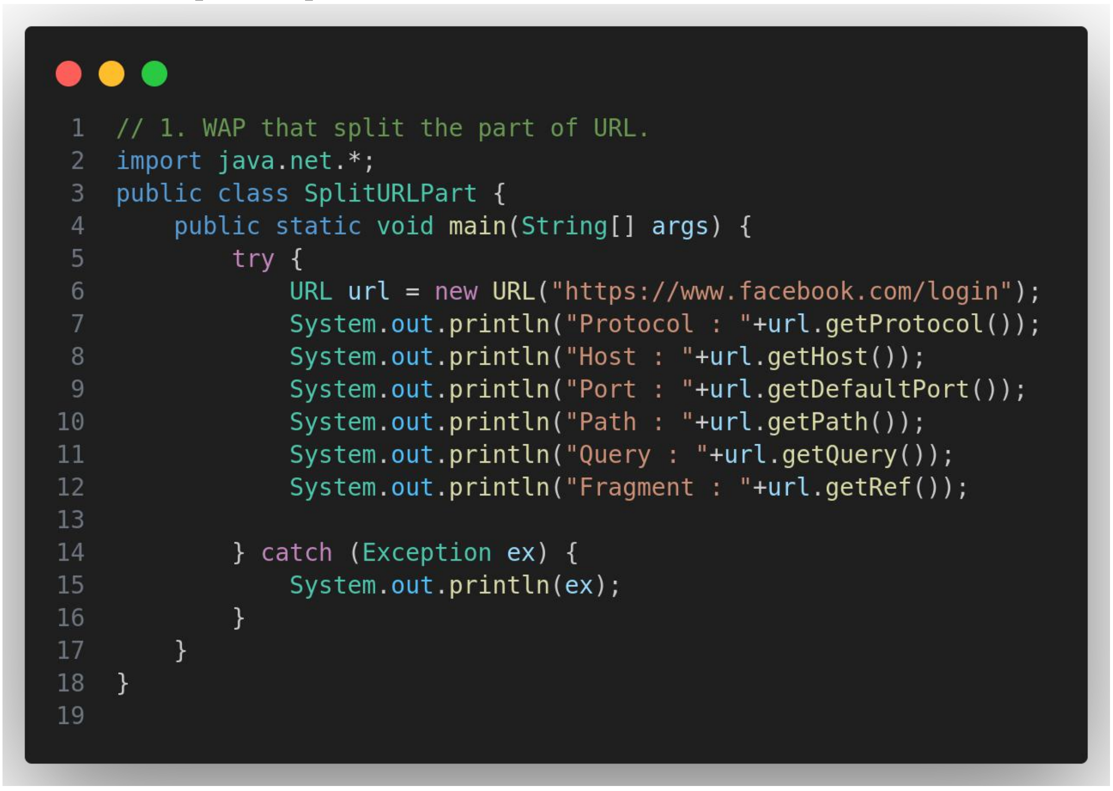
13. WAP to process Web Server log file.



```java
// 13. WAP to process Web Server log file.
import java.io.*;
import java.net.*;
public class ProcessWebServerLogFile {
    public static void main(String[] args) {
        String file="logfiles.txt";
        try{
            FileInputStream fin = new FileInputStream(file);
            Reader in = new InputStreamReader(fin);
            BufferedReader bin = new BufferedReader(in);
            while(bin.readLine()!=null) {
                // separate out the IP address
                String entry = bin.readLine();
                int index = entry.indexOf(' ');
                String ip = entry.substring(0, index);
                String theRest = entry.substring(index);
                // Ask DNS for the hostname and print it out
                try {
                    InetAddress address = InetAddress.getByName(ip);
                    System.out.println(address.getHostName()+ theRest);
                } catch (UnknownHostException ex) {
                    System.out.println(ex);
                }
            }
        } catch (IOException ex) {
            System.out.println("Exception: " + ex);
        }
    }
}
```

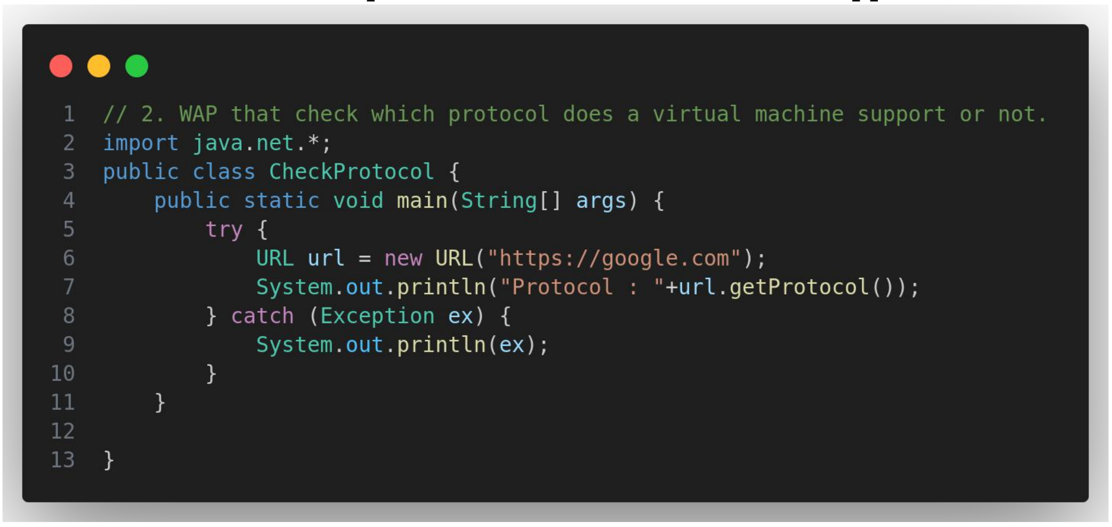
Unit - 3

1. WAP that split the part of URL.



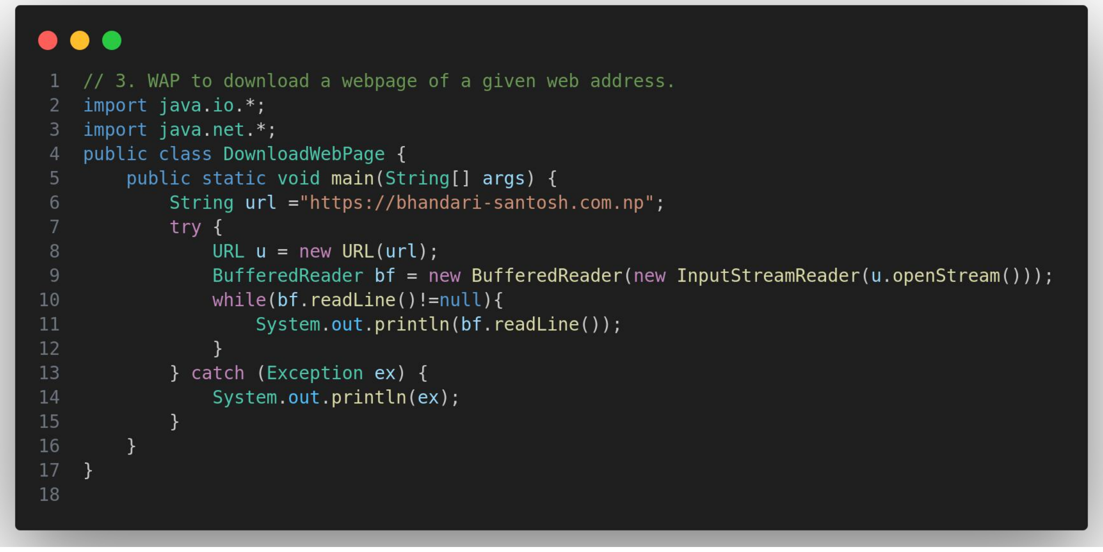
```java
// 1. WAP that split the part of URL.
import java.net.*;
public class SplitURLPart {
    public static void main(String[] args) {
        try {
            URL url = new URL("https://www.facebook.com/login");
            System.out.println("Protocol : "+url.getProtocol());
            System.out.println("Host : "+url.getHost());
            System.out.println("Port : "+url.getDefaultPort());
            System.out.println("Path : "+url.getPath());
            System.out.println("Query : "+url.getQuery());
            System.out.println("Fragment : "+url.getRef());

        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

2. WAP that check which protocol does a virtual machine support or not.



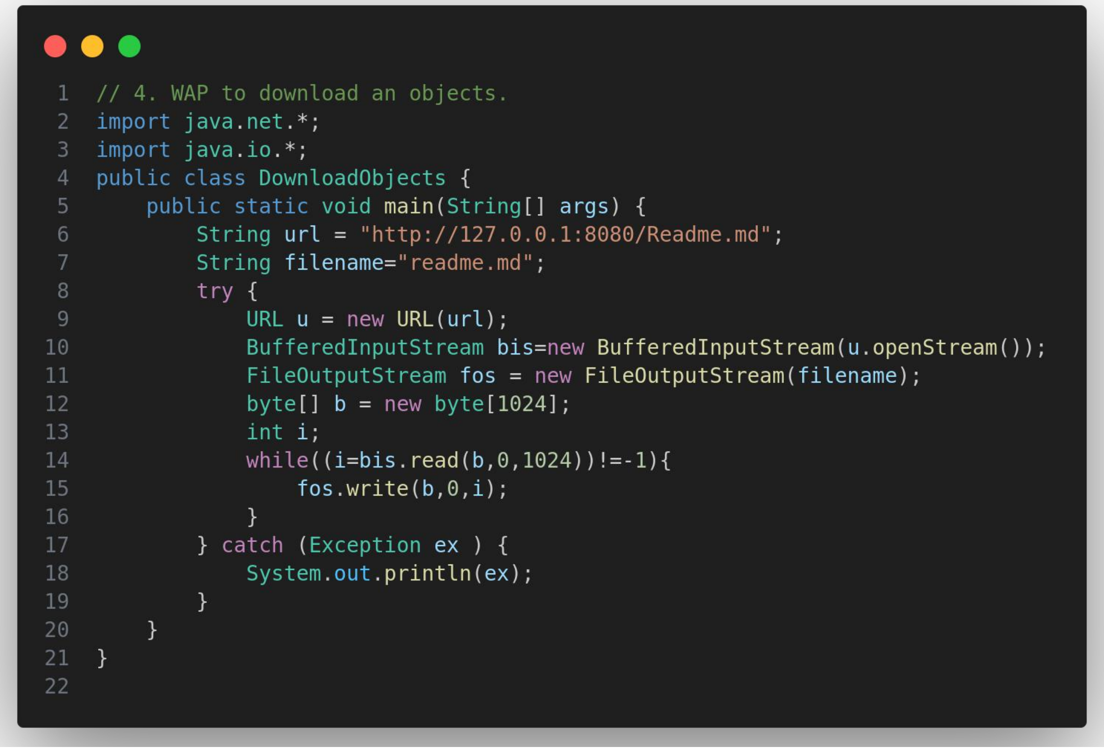
```java
// 2. WAP that check which protocol does a virtual machine support or not.
import java.net.*;
public class CheckProtocol {
    public static void main(String[] args) {
        try {
            URL url = new URL("https://google.com");
            System.out.println("Protocol : "+url.getProtocol());
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

}
```

3. WAP to download a webpage of a given web address.



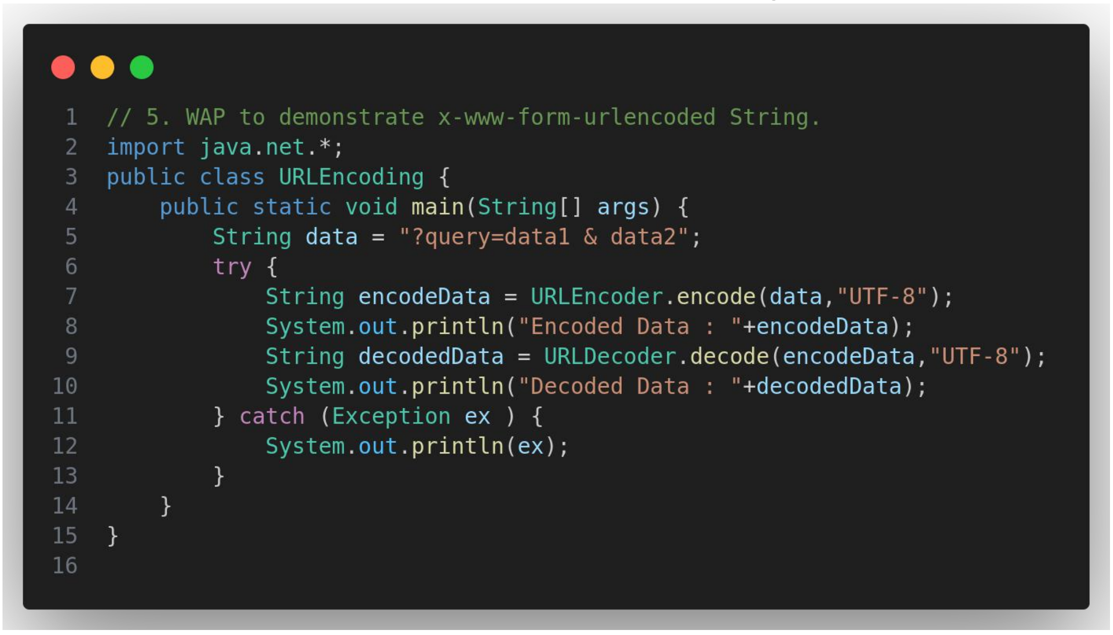
```java
// 3. WAP to download a webpage of a given web address.
import java.io.*;
import java.net.*;
public class DownloadWebPage {
    public static void main(String[] args) {
        String url ="https://bhandari-santosh.com.np";
        try {
            URL u = new URL(url);
            BufferedReader bf = new BufferedReader(new InputStreamReader(u.openStream()));
            while(bf.readLine()!=null){
                System.out.println(bf.readLine());
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

4. WAP to download an objects.



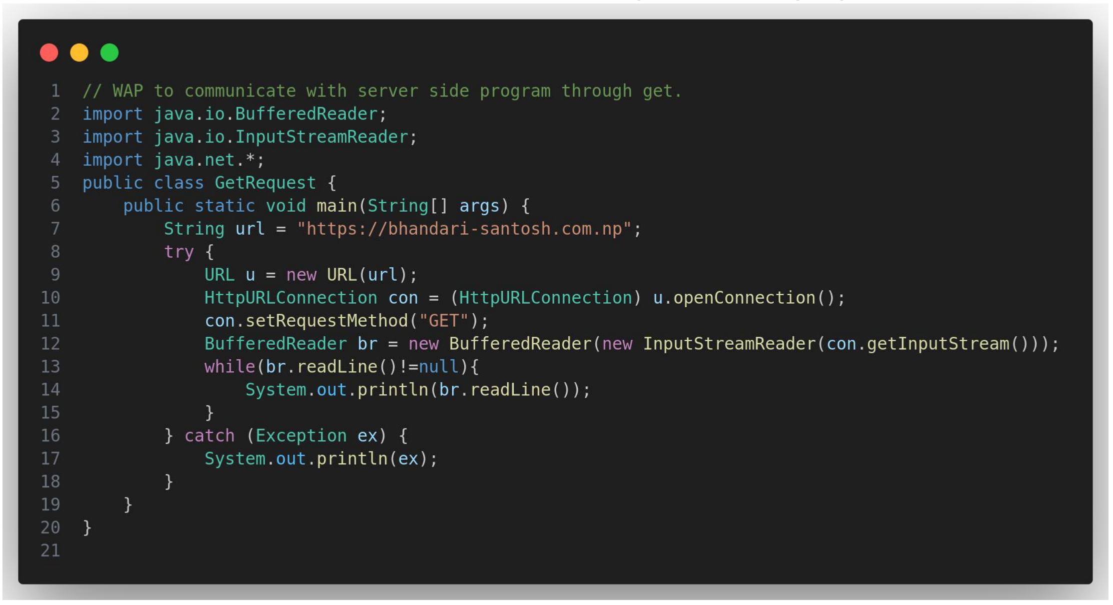
```java
// 4. WAP to download an objects.
import java.net.*;
import java.io.*;
public class DownloadObjects {
    public static void main(String[] args) {
        String url = "http://127.0.0.1:8080/Readme.md";
        String filename="readme.md";
        try {
            URL u = new URL(url);
            BufferedInputStream bis=new BufferedInputStream(u.openStream());
            FileOutputStream fos = new FileOutputStream(filename);
            byte[] b = new byte[1024];
            int i;
            while((i=bis.read(b,0,1024))!=-1){
                fos.write(b,0,i);
            }
        } catch (Exception ex ) {
            System.out.println(ex);
        }
    }
}
```

5. WAP to demonstrate x-www-form-urlencoded String.



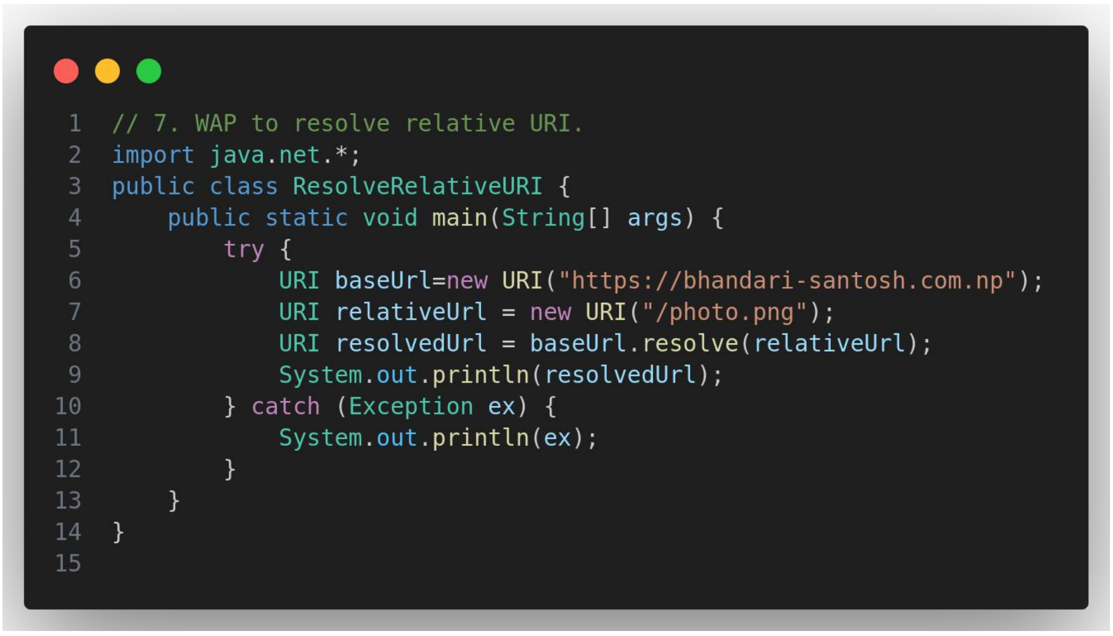
```java
// 5. WAP to demonstrate x-www-form-urlencoded String.
import java.net.*;
public class URLEncoding {
    public static void main(String[] args) {
        String data = "?query=data1 & data2";
        try {
            String encodeData = URLEncoder.encode(data,"UTF-8");
            System.out.println("Encoded Data : "+encodeData);
            String decodedData = URLDecoder.decode(encodeData,"UTF-8");
            System.out.println("Decoded Data : "+decodedData);
        } catch (Exception ex ) {
            System.out.println(ex);
        }
    }
}

```

6. WAP to communicate with server side program through get.



```java
// WAP to communicate with server side program through get.
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.*;
public class GetRequest {
    public static void main(String[] args) {
        String url = "https://bhandari-santosh.com.np";
        try {
            URL u = new URL(url);
            HttpURLConnection con = (HttpURLConnection) u.openConnection();
            con.setRequestMethod("GET");
            BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream()));
            while(br.readLine()!=null){
                System.out.println(br.readLine());
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

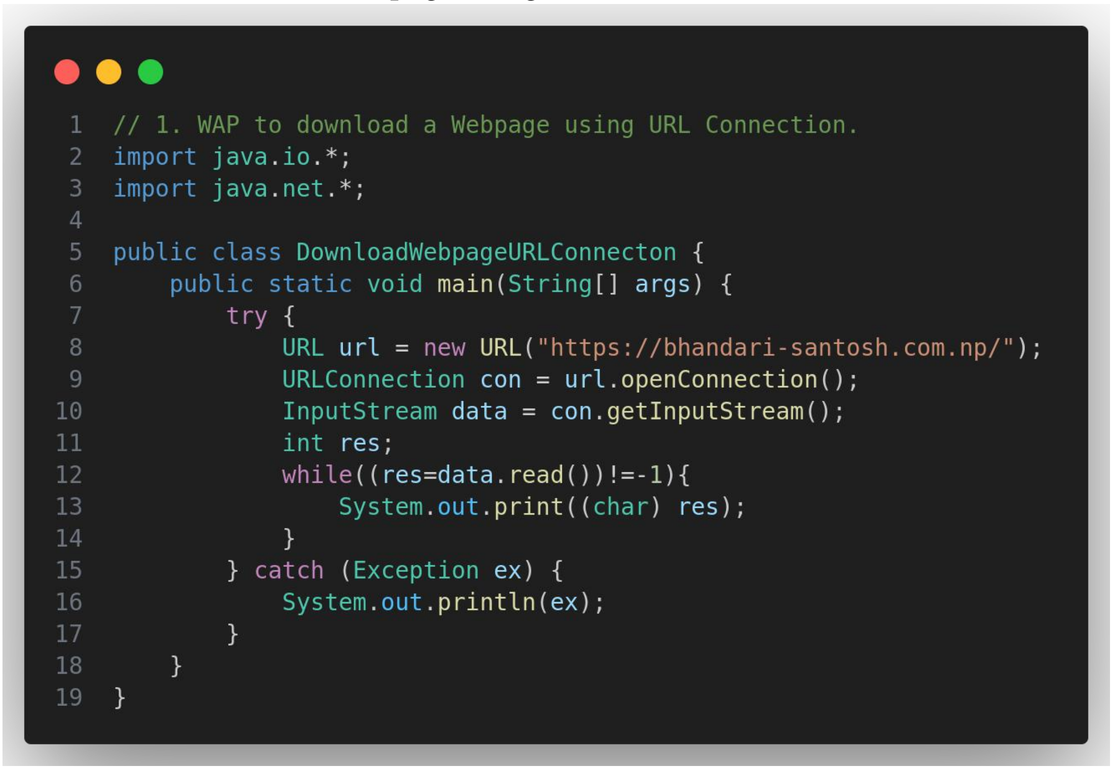
7. WAP to resolve relative URI.



```java
// 7. WAP to resolve relative URI.
import java.net.*;
public class ResolveRelativeURI {
    public static void main(String[] args) {
        try {
            URI baseUrl=new URI("https://bhandari-santosh.com.np");
            URI relativeUrl = new URI("/photo.png");
            URI resolvedUrl = baseUrl.resolve(relativeUrl);
            System.out.println(resolvedUrl);
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

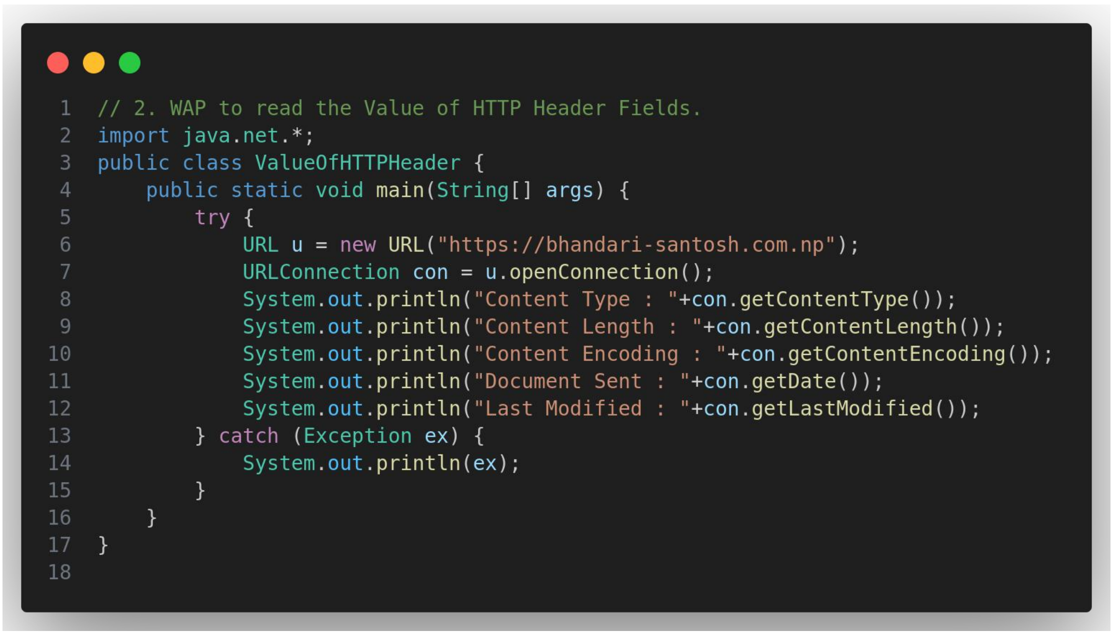
Unit - 5

1. WAP to download a Webpage using URL Connection.



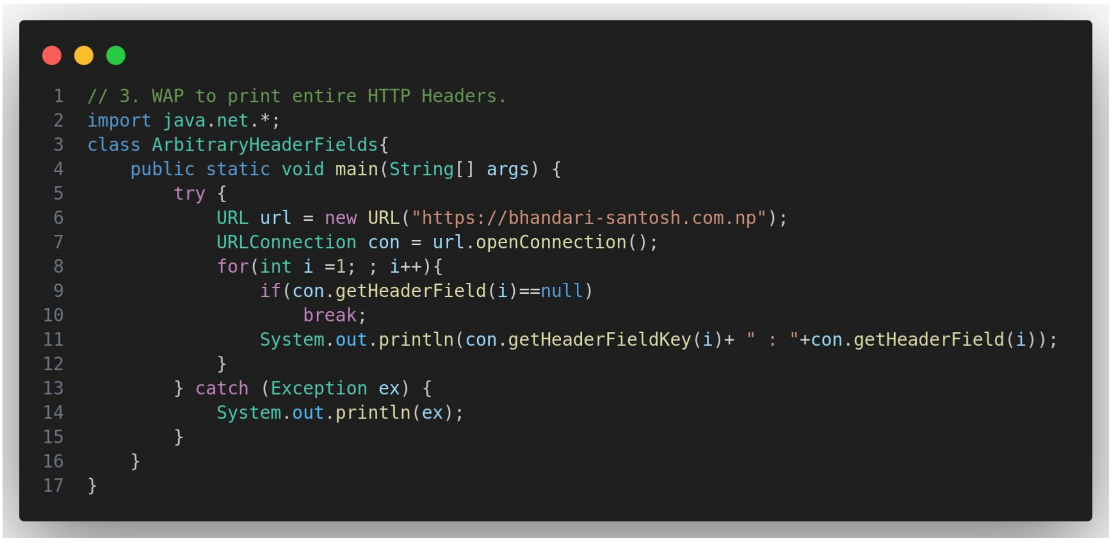
```java
// 1. WAP to download a Webpage using URL Connection.
import java.io.*;
import java.net.*;

public class DownloadWebpageURLConnecton {
    public static void main(String[] args) {
        try {
            URL url = new URL("https://bhandari-santosh.com.np/");
            URLConnection con = url.openConnection();
            InputStream data = con.getInputStream();
            int res;
            while((res=data.read())!=-1){
                System.out.print((char) res);
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

2. WAP to read the Value of HTTP Header Fields.



```java
// 2. WAP to read the Value of HTTP Header Fields.
import java.net.*;
public class ValueOfHTTPHeader {
    public static void main(String[] args) {
        try {
            URL u = new URL("https://bhandari-santosh.com.np");
            URLConnection con = u.openConnection();
            System.out.println("Content Type : "+con.getContentType());
            System.out.println("Content Length : "+con.getContentLength());
            System.out.println("Content Encoding : "+con.getContentEncoding());
            System.out.println("Document Sent : "+con.getDate());
            System.out.println("Last Modified : "+con.getLastModified());
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

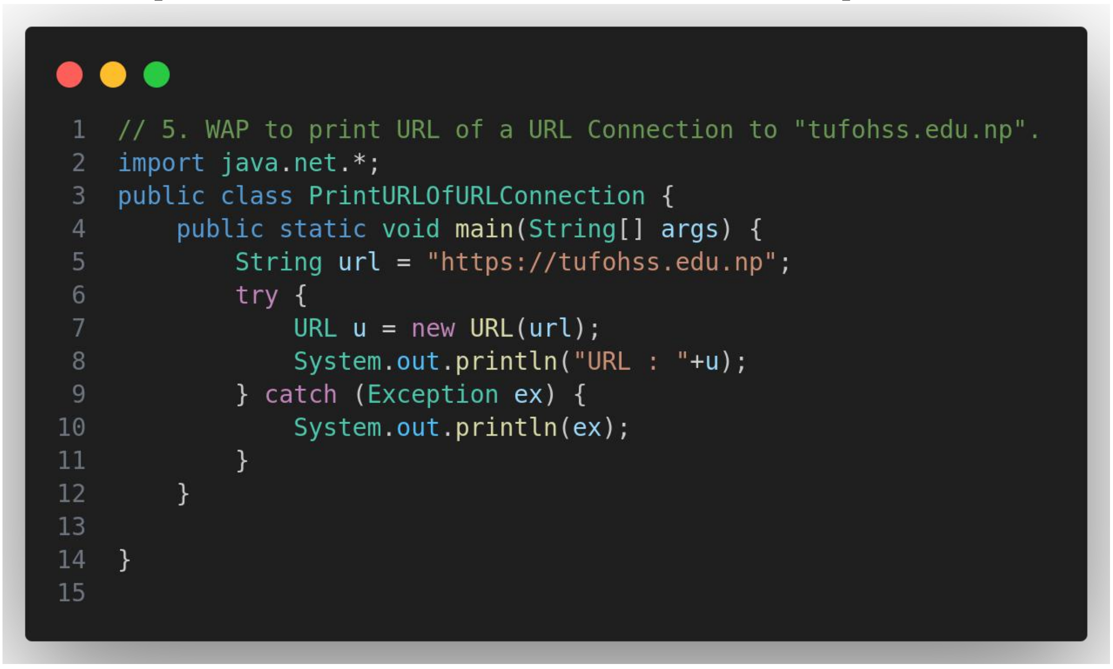
3. WAP to print entire HTTP Headers.



```java
// 3. WAP to print entire HTTP Headers.
import java.net.*;
class ArbitraryHeaderFields{
    public static void main(String[] args) {
        try {
            URL url = new URL("https://bhandari-santosh.com.np");
            URLConnection con = url.openConnection();
            for(int i =1; ; i++){
                if(con.getHeaderField(i)==null)
                    break;
                System.out.println(con.getHeaderFieldKey(i)+ " : "+con.getHeaderField(i));
            }
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

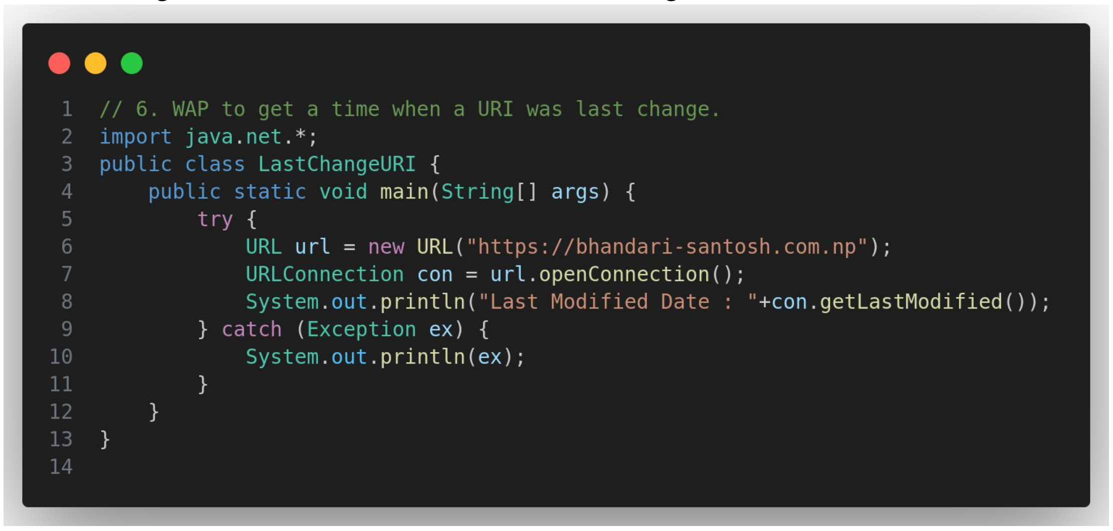
4. WAP for HTTP request Method. - Not Done

5. WAP to print URL of a URL Connection to "tufohss.edu.np".



```java
// 5. WAP to print URL of a URL Connection to "tufohss.edu.np".
import java.net.*;
public class PrintURLOfURLConnection {
    public static void main(String[] args) {
        String url = "https://tufohss.edu.np";
        try {
            URL u = new URL(url);
            System.out.println("URL : "+u);
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

}

```

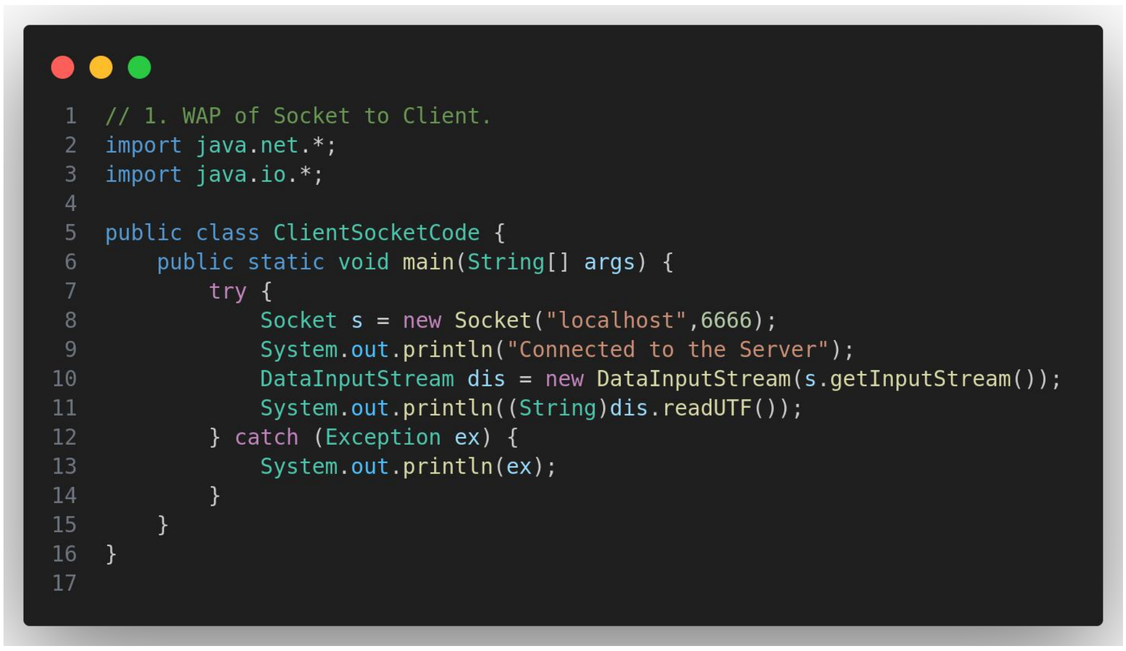
6. WAP to get a time when a URI was last change.



```java
// 6. WAP to get a time when a URI was last change.
import java.net.*;
public class LastChangeURI {
    public static void main(String[] args) {
        try {
            URL url = new URL("https://bhandari-santosh.com.np");
            URLConnection con = url.openConnection();
            System.out.println("Last Modified Date : "+con.getLastModified());
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

Unit - 6

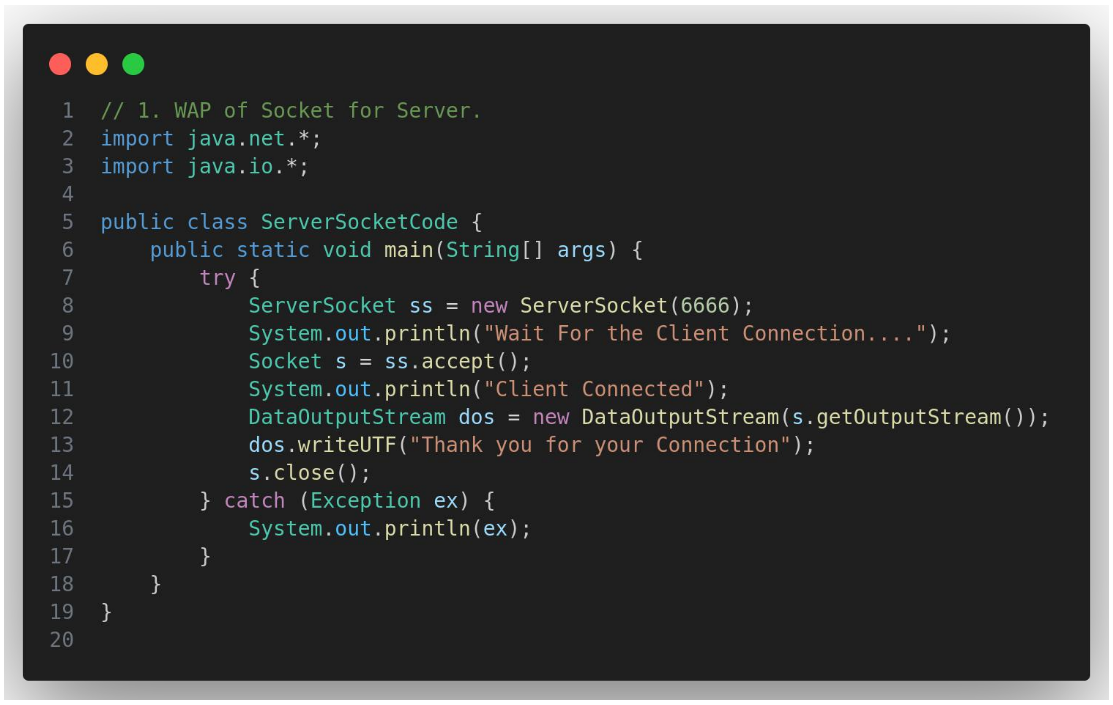
1. WAP of Socket to Client.



```java
// 1. WAP of Socket to Client.
import java.net.*;
import java.io.*;

public class ClientSocketCode {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("localhost",6666);
            System.out.println("Connected to the Server");
            DataInputStream dis = new DataInputStream(s.getInputStream());
            System.out.println((String)dis.readUTF());
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}
```

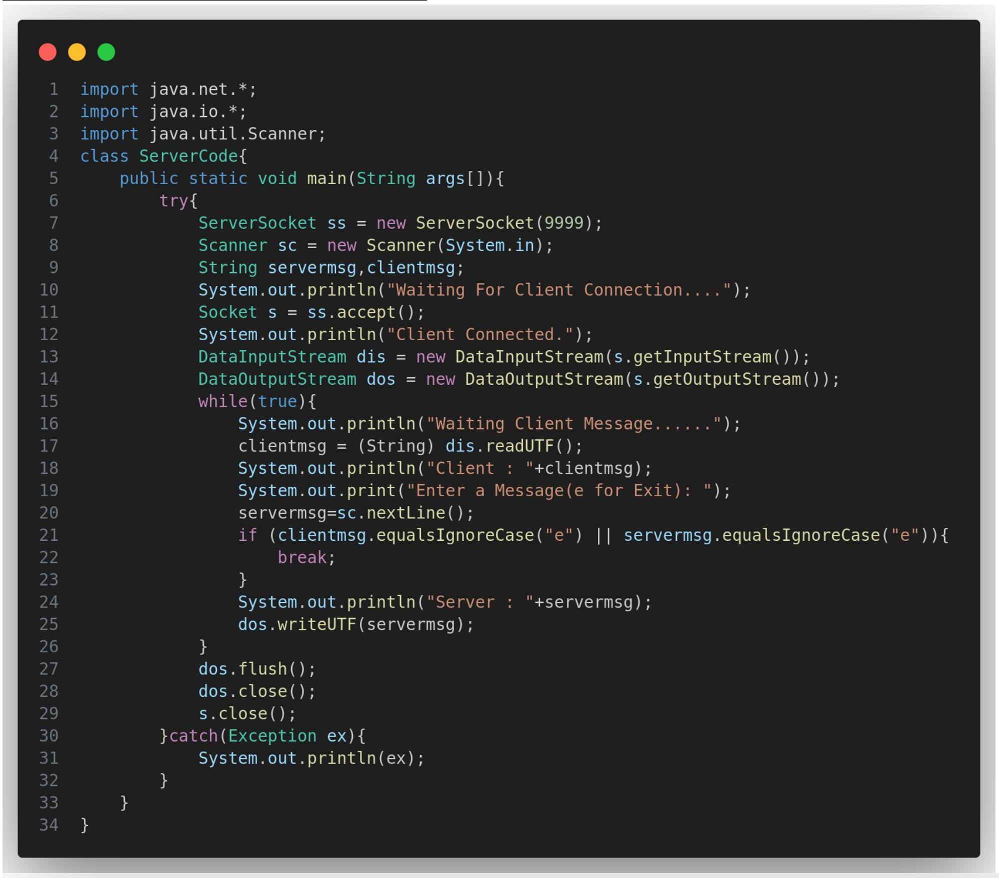
Unit - 7

1. WAP of Socket for Server.

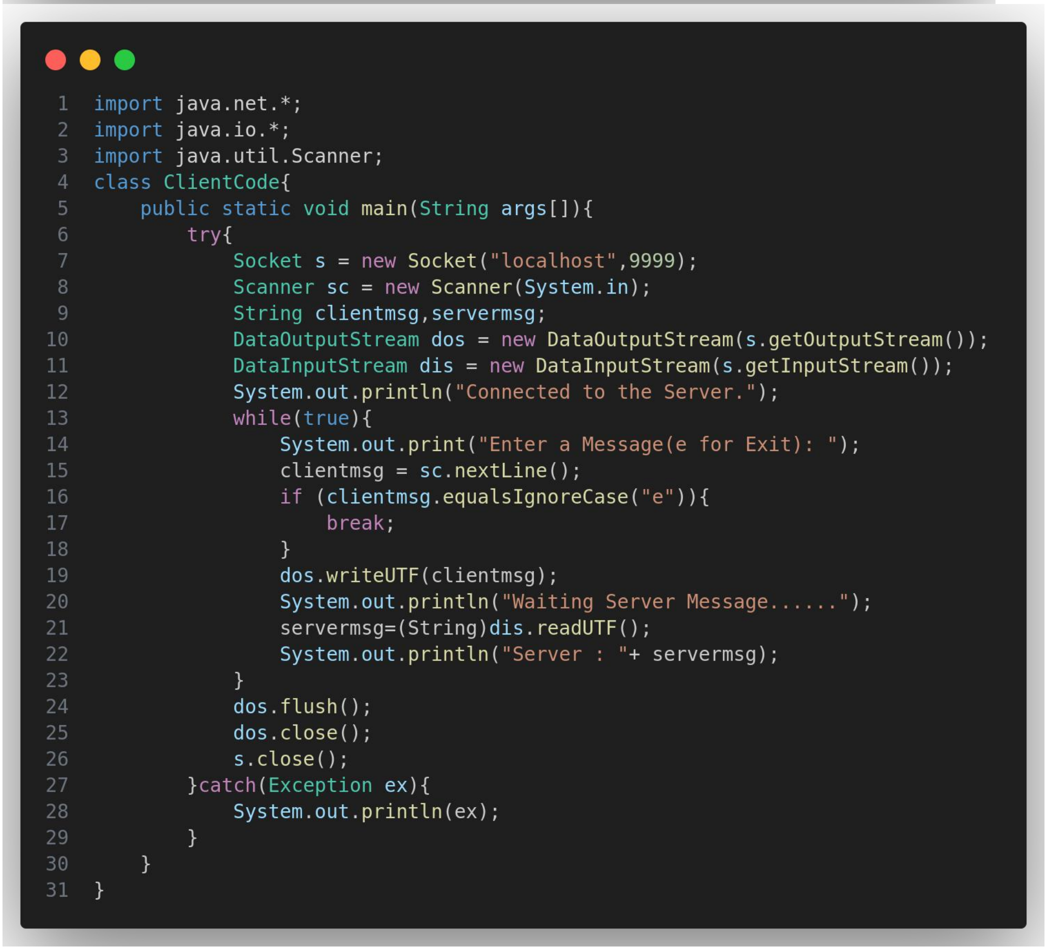


```java
// 1. WAP of Socket for Server.
import java.net.*;
import java.io.*;

public class ServerSocketCode {
    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(6666);
            System.out.println("Wait For the Client Connection....");
            Socket s = ss.accept();
            System.out.println("Client Connected");
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            dos.writeUTF("Thank you for your Connection");
            s.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }
}

```

Client Server Communication



```java
import java.net.*;
import java.io.*;
import java.util.Scanner;
class ServerCode{
    public static void main(String args[]){
        try{
            ServerSocket ss = new ServerSocket(9999);
            Scanner sc = new Scanner(System.in);
            String servermsg,clientmsg;
            System.out.println("Waiting For Client Connection....");
            Socket s = ss.accept();
            System.out.println("Client Connected.");
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            while(true){
                System.out.println("Waiting Client Message......");
                clientmsg = (String) dis.readUTF();
                System.out.println("Client : "+clientmsg);
                System.out.print("Enter a Message(e for Exit): ");
                servermsg=sc.nextLine();
                if (clientmsg.equalsIgnoreCase("e") || servermsg.equalsIgnoreCase("e")){
                    break;
                }
                System.out.println("Server : "+servermsg);
                dos.writeUTF(servermsg);
            }
            dos.flush();
            dos.close();
            s.close();
        }catch(Exception ex){
            System.out.println(ex);
        }
    }
}
```



```java
import java.net.*;
import java.io.*;
import java.util.Scanner;
class ClientCode{
    public static void main(String args[]){
        try{
            Socket s = new Socket("localhost",9999);
            Scanner sc = new Scanner(System.in);
            String clientmsg,servermsg;
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            DataInputStream dis = new DataInputStream(s.getInputStream());
            System.out.println("Connected to the Server.");
            while(true){
                System.out.print("Enter a Message(e for Exit): ");
                clientmsg = sc.nextLine();
                if (clientmsg.equalsIgnoreCase("e")){
                    break;
                }
                dos.writeUTF(clientmsg);
                System.out.println("Waiting Server Message......");
                servermsg=(String)dis.readUTF();
                System.out.println("Server : "+ servermsg);
            }
            dos.flush();
            dos.close();
            s.close();
        }catch(Exception ex){
            System.out.println(ex);
        }
    }
}
```