**LAB – 3: URL Connection Class**

1. A program to download a webpage with URL Connection.

import java.io.\*;

import java.net.\*;

public class SourceViewer2{

public static void main (String[] args) {

 if (args.length > 0) {

 try {

 // Open the URLConnection for reading

 URL u = new URL(args[0]);

 URLConnection uc = u.openConnection();

 try (InputStream raw = uc.getInputStream()) { // autoclose

 InputStream buffer = new BufferedInputStream(raw);

 // chain the InputStream to a Reader

 Reader reader = new InputStreamReader(buffer);

 int c;

 while ((c = reader.read()) != -1) {

 System.out.print((char) c);

 }

 }

 } catch (MalformedURLException ex) {

 System.err.println(args[0] + " is not a parseable URL");

 } catch (IOException ex) {

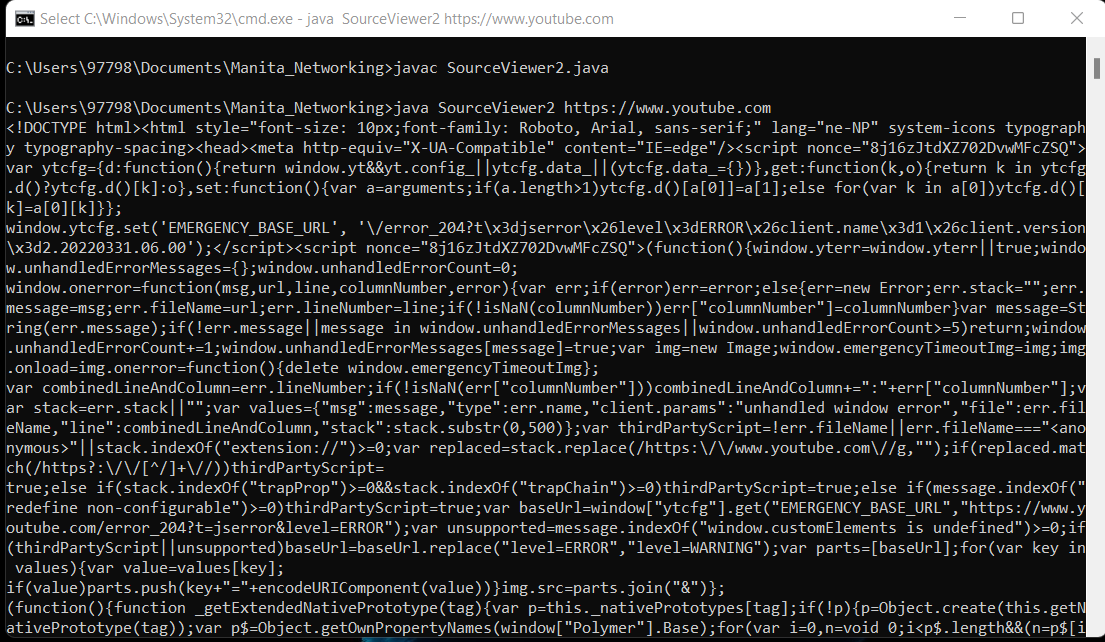
 System.err.println(ex);

 } }

 }

}

**Output:**

****

1. A program that reads URLs from the command line and uses these six methods to print their content type, content length, content encoding, date of last modification, expiration date, and current date.

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class HeaderViewer {

 public static void main(String[] args) {

 for (int i = 0; i < args.length; i++) {

 try {

 URL u = new URL(args[0]);

 URLConnection uc = u.openConnection();

 System.out.println("Content-type: " + uc.getContentType());

 if (uc.getContentEncoding() != null) {

 System.out.println("Content-encoding:  + uc.getContentEncoding());

 }

 if (uc.getDate() != 0) {

 System.out.println("Date: " + new Date(uc.getDate()));

 }

 if (uc.getLastModified() != 0) {

 System.out.println("Last modified: " + new Date(uc.getLastModified()));

 }

 if (uc.getExpiration() != 0) {

 System.out.println("Expiration date: " + new Date(uc.getExpiration()));

 }

 if (uc.getContentLength() != -1) {

 System.out.println("Content-length: " + uc.getContentLength()); }

 } catch (MalformedURLException ex) {

 System.err.println(args[i] + " is not a URL I understand");

 } catch (IOException ex) {

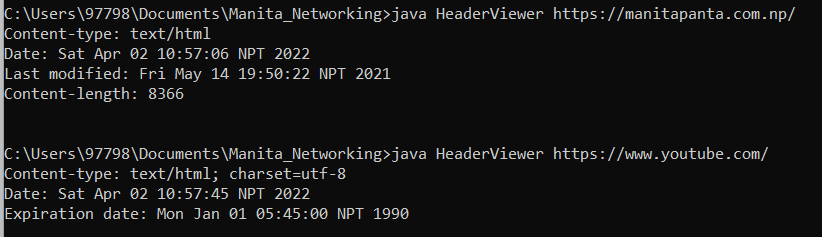
 System.err.println(ex);

 }

 System.out.println();

 } } }

**Output:**

****

1. A program to print entire HTTP Header.

import java.io.\*;

import java.net.\*;

public class AllHeaders {

 public static void main(String[] args) {

 for (int i = 0; i < args.length; i++) {

 try {

 URL u = new URL(args[i]);

 URLConnection uc = u.openConnection();

 for (int j = 1; ; j++) {

 String header = uc.getHeaderField(j);

 if (header == null) break;

 System.out.println(uc.getHeaderFieldKey(j) + ": " + header); }

 } catch (MalformedURLException ex) {

 System.err.println(args[i] + " is not a URL I understand.");

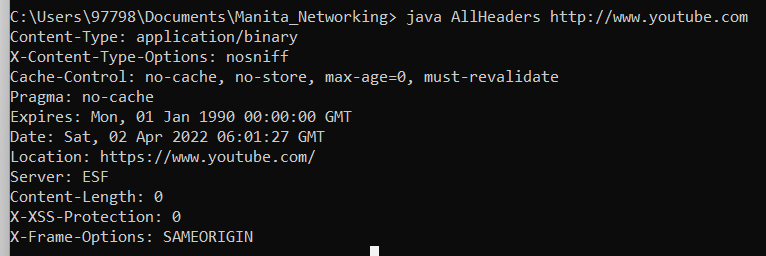
 } catch (IOException ex) {

 System.err.println(ex); }

 System.out.println();

 } } }

Output:



1. A program that prints the default value of ifModifiedSince, sets its value to 24 hours ago, and prints the new value. It then downloads and displays the document—but only if it’s been modified in the last 24 hours.

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class Last24 {

 public static void main (String[] args) {

 // Initialize a Date object with the current date and time

 Date today = new Date();

 long millisecondsPerDay = 24 \* 60 \* 60 \* 1000;

 for (int i = 0; i < args.length; i++) {

 try {

 URL u = new URL(args[i]);

 URLConnection uc = u.openConnection();

 System.out.println("Original if modified since: " + new Date(uc.getIfModifiedSince()));

 uc.setIfModifiedSince((new Date(today.getTime() - millisecondsPerDay)).getTime());

 System.out.println("Will retrieve file if it's modified since "+ new Date(uc.getIfModifiedSince()));

 try (InputStream in = new BufferedInputStream(uc.getInputStream())) {

Reader r = new InputStreamReader(in);

 int c;

 while ((c = r.read()) != -1) {

 System.out.print((char) c); }

 System.out.println(); }

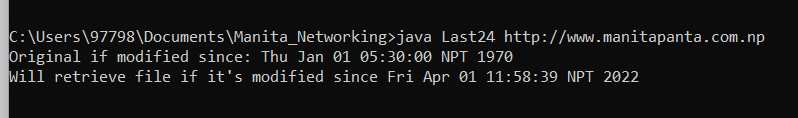
 } catch (IOException ex) {

 System.err.println(ex);

 }

 } } }

**Output:**

****

1. A program to get the time when the URL was last changed.

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class LastModified {

 public static void main(String[] args) {

 for (int i = 0; i < args.length; i++) {

 try {

 URL u = new URL(args[i]);

 HttpURLConnection http = (HttpURLConnection) u.openConnection();

 http.setRequestMethod("HEAD");

 System.out.println(u + " was last modified at " + new Date(http.getLastModified()));

 } catch (MalformedURLException ex) {

 System.err.println(args[i] + " is not a URL I understand");

 } catch (IOException ex) {

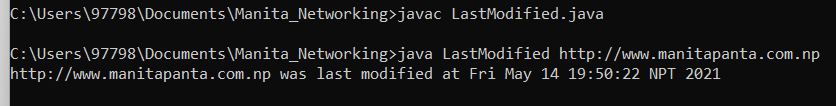
 System.err.println(ex); }

 System.out.println(); }

 }

}

**Output:**



1. A program that return response code and response message.

import java.io.\*;

import java.net.\*;

public class SourceViewer3 {

 public static void main (String[] args) {

 for (int i = 0; i < args.length; i++) {

 try {

  URL u = new URL(args[i]);

 HttpURLConnection uc = (HttpURLConnection) u.openConnection();

 int code = uc.getResponseCode();

 String response = uc.getResponseMessage();

 System.out.println("HTTP/1.x " + code + " " + response);

for (int j = 1; ; j++) {

 String header = uc.getHeaderField(j);

 String key = uc.getHeaderFieldKey(j);

 if (header == null || key == null) break;

 System.out.println(uc.getHeaderFieldKey(j) + ": " + header); }

 System.out.println();

 try (InputStream in = new BufferedInputStream(uc.getInputStream())) {

 Reader r = new InputStreamReader(in);

 int c;

 while ((c = r.read()) != -1) {

 System.out.print((char) c); } }

 } catch (MalformedURLException ex) {

 System.err.println(args[0] + " is not a parseable URL");

 } catch (IOException ex) {

 System.err.println(ex);

 }

 }

 }

}

**Output:**

