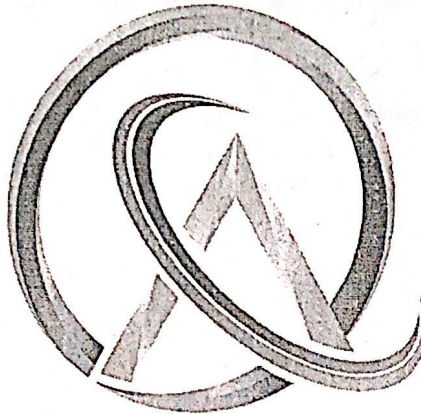


# INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

KALANKI, KATHMANDU

(AFFILIATED TO TRIBHUVAN UNIVERSITY)



ADVANCED COLLEGE  
OF ENGINEERING & MANAGEMENT

## LAB REPORT

SUBJECT : Network Programming

LAB NO : 2

### SUBMITTED BY:

NAME : Anish Shrestha

ROLL NO: 1

DATE : 20 82/09/30

### SUBMITTED TO:

BCA Department

*Q. Anish Shrestha*  
14/09/2020

# TITLE: URLs AND URIs

## OBJECTIVES:

- To understand the structure of URL and how to parse its components.
- To learn how to programmatically download web pages and objects using the URL class.
- To implement Relative URI resolution.

## THEORY:

### URL (Uniform Resource Locator)

- The `java.net.URL` class represents a Uniform Resource Locator, which is a pointer to a "resource" on the World Wide Web.
- The URL class provides methods to parse the URL string and to open a connection to the resource to read data.

### URI (Uniform Resource Identifier)

- The `java.net.URI` class represents a Uniform Resource Identifier.
- A URI is a sequence of characters that identifies a logical or physical resource.
- The URI class is often used for parsing and manipulating identifier strings before converting them to URLs.

## SOURCE CODE:

```
package lab2;

import java.net.*;
import java.nio.*;
import java.io.*;

public class Solution{

    public static void main(String[] args){

        question 1();
        question 2();
        question 3();
        question 4();

    }
```

1/1. Program to split parts of URL

```
public static void question1(){
```

```
    String url = "https://www.example.com:8080/path/to/
    resource?query=value&another=a.value
    #section-id";
```

```
try{
```

```
    URL urlObj = new URL(url);
```

```
    System.out.println("Protocol:" + urlObj.getProtocol());
```

```
    System.out.println("Host:" + urlObj.getHost());
```

```
    System.out.println("Path:" + urlObj.getPath());
```

```
    System.out.println("Port:" + urlObj.getPort());
```

```

        System.out.println ("Query:" + urlObj.getQuery());
        System.out.println ("Fragment:" + urlObj.getRef());
    } catch (MalformedURLException e) {
        System.out.println (e.getMessage());
    }
}

```

```

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:f
Portocol: https
Host: www.example.com
Path: /path/to/resource
Port: 8080
Query: query_param=value&another_param=another_value
Fragment: section-id

Process finished with exit code 0

```

112 Program to download web page

```

public static void question2() {

```

```

    String acem = "https://acem.edu.pp/";

```

```

    try {

```

```

        URL url = new URL (acem);

```

```

        InputStream istream = url.openStream(istream);

```

```

        Reader r = new InputStreamReader (istream);

```

```

        int c;

```

```

        while ( (c=r.read()) != -1) {

```

```

            System.out.println ( (char) c);

```

```

            if (c == '>') System.out.println();

```

```

        }

```

```

    } catch (IOException e) {

```

```

        System.out.println (e.getMessage());
    }
}

```

```

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\downloadssss\Intell
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>
Home | Advanced College of Engineering and Management</title>
<link rel="icon" href="https://acem.edu.np/assets/images/xfavicon.png.pagespee
<link rel="shortcut icon" href="https://acem.edu.np/assets/images/xfavicon.png
<meta property="og:title" content="Home"/>
<meta property="og:type" content="Website"/>
<meta property="og:url" content="https://acem.edu.np/">
<meta property="og:image" content="https://acem.edu.np/assets/images/default.j
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;700
<link href="https://acem.edu.np/assets/css/bootstrap.min.css" rel="stylesheet"
<link href="https://acem.edu.np/assets/css/style.min.css?v=035438" rel="styles
<link href="https://acem.edu.np/assets/css/fa-all.min.css" rel="stylesheet">
<link href="https://acem.edu.np/plugins/owlcarousel/assets/owl.carousel.min.cs
<link href="https://acem.edu.np/plugins/owlcarousel/assets/owl.theme.default.m
<link href="https://acem.edu.np/plugins/uikit/css/uikit.min.css" rel="styleshe
<script src="https://acem.edu.np/plugins/jquery/jquery.min.js.pagespeed.jm.A8t
</script>

```

113 to download object

```
public static void question3() {
```

```
    String imgUrl = "https://acem.edu.np/uploads/.../img.webp";
```

```
    String destination = "D:\\Applications\\image.jpg";
```

```
    try { InputStream in = new URL(imgUrl).openStream();
```

```
        Files.copy(in, Paths.get(destination), StandardCopyOption.
REPLACE_EXISTING);
```

```
        System.out.println("Image downloaded successfully: " +
destination);
```

```
    } catch (IOException e) {
```

```
        System.out.println(e.getMessage());
```

```
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:
Image downloaded successfully: D:\Applications\image.j
Process finished with exit code 0
```

Name	Date modified
✓ Today	
image	1/11/2026 3:56 PM
✓ Earlier this month	

114 restoring relative URI

```
public static void question4(){
```

```
try{
```

```
URI base = new URI("https://example.com/folder/");
```

```
URI relativeUri = new URI("images/photo.jpg");
```

```
URI resolvedUri = base.resolve(relativeUri);
```

```
System.out.println("Base URI:" + base);
```

```
System.out.println("Relative URI:" + relativeUri);
```

```
System.out.println("Resolved URI:" + resolvedUri);
```

```
} catch (URISyntaxException e){
```

```
System.out.println(e.getMessage());
```

```
}
```

```
}
```

```
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\downloadss:
Base URI: https://example.com/folder/
Relative URI: images/photo.jpg
Resolved URI: https://example.com/folder/images/photo.jpg
Process finished with exit code 0
```

## CONCLUSION:

In this lab, we focused on handling URLs and URIs within the Java environment. We successfully implemented parsing URL, downloading web pages, objects and also resolving relative paths with the help of URL class, URI class and File handling.

