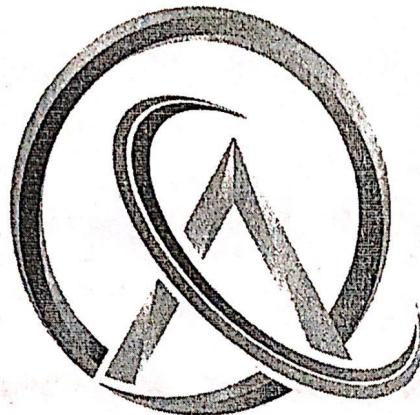
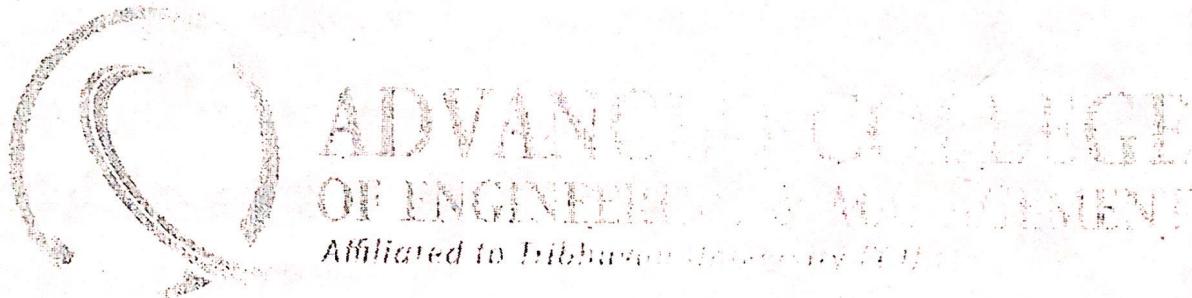


INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT
KALANKI, KATHMANDU
(AFFILIATED TO TRIBHUVAN UNIVERSITY)



**ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT**



**ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT**
Affiliated to Tribhuvan University, Kathmandu

LAB REPORT

SUBJECT : Network Programming
LAB NO : 3

SUBMITTED BY:

NAME : Anish Shrestha
ROLL NO: 1
DATE : 2082/09/30

SUBMITTED TO:

BCA Department
Qamrul Hasan

TITLE: HTTP AND COOKIE HANDLING

OBJECTIVES:

- To understand how to connect to local HTTP server using Java.
- To understand the concept of HTTP Cookies and their role in session management.
- To learn to implement the CookieHandler and CookieManager classes in Java.

THEORY:

HTTP (Hypertext Transfer Protocol)

- It is the protocol used by the World Wide Web that defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response.
- It is stateless.

CookieHandler

- The `java.net.CookieHandler` is an abstract class that provides a callback mechanism to manage cookies within the HTTP state management policy.
- It intercepts requests to add appropriate cookies and intercepts responses to save new cookies.

Cookie Manager

→ The `java.net.CookieManager` is a concrete implementation of `CookieHandler`. It separates the logic storage of cookie from the policy surrounding accepting and rejecting cookies.

SOURCE CODE:

```
package lab3;
import java.*;
public class Solution{
    public static void main(String[] args){
        question1();
        question2();
    }
}

// program to connect to server running on localhost
private static void question1(){
    String urlString="http://127.0.0.1:8080/index.html";
    System.out.println("Attempting to connect to :");
    try{
        URL url= new URL(urlString);
```

```

HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("GET");
int responseCode = conn.getResponseCode();
System.out.println("Server Response Code :" + responseCode);
if (responseCode == 200) {
    BufferedReader in = new BufferedReader(new
        InputStreamReader(conn.getInputStream()));
    String inputLine;
    int count = 0;
    while ((inputLine = in.readLine()) != null && count < 5)
    {
        System.out.println(inputLine);
        count++;
    }
    in.close();
}
} catch (ConnectException e) {
    System.out.println(e.getMessage());
}
}

```

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:D:\dow
 Attempting to connect to: http://127.0.0.1:8080/index.html
 Server Response Code: 200
 <!DOCTYPE html>
 <html lang="en">
 <head>
 <meta charset="UTF-8">
 Process finished with exit code 0

112 To retrieve cookie from server

```
public static void question2(){
    CookieManager cm = new CookieManager();
    CookieHandler.setDefault(cm);

    try {
        URL url = new URL("http://httpbin.org/cookies/set?");
        myCookie = "Java Lab & session=12345";
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.connect();

        Map<String, List<String>> headerFields = conn.getHeaderFields();
        List<String> cookieHeader = headerFields.get("Set-Cookie");
        if (cookieHeader != null) {
            for (String c : cookieHeader)
                cm.getCookieStore().add(null, HttpCookie.parse(cookieHeader.get(0)));
        }

        System.out.println("Cookies: " + cm.getCookieStore().getCookies());
        conn.disconnect();
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    }
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent  
Cookies: [myCookie=JavaLab, session=12345]  
Process finished with exit code 0.
```

CONCLUSION:

In this lab, we successfully explored client-server interaction and state management in Java. We implemented a client program to connect to a local HTTP server running on port 8080. We also utilized ~~CookieHandler~~ and ~~CookieManager~~ classes to handle HTTP cookies.