EXP.NO: 3   Write a Servlet program that prints "Hello, World!"

when accessed through a browser.

AIM:

To write a basic Servlet program that demonstrates the use of Java Servlets by printing

"Hello, World!" when accessed through a browser.

ALGORITHM:

1. Input:

▪ The user makes an HTTP request (usually by entering a URL) through a browser.

2. Servlet Program Flow:

1. Create a Servlet Class:

o Define a Java class that extends the HttpServlet class.

o Override the doGet() method, which handles HTTP GET requests from the

client (browser).

2. Handle HTTP Request:

o In the doGet() method:

▪ Get the PrintWriter object to write the response to the browser.

▪ Set the response content type to text/html.

▪ Write the "Hello, World!" message to the response.

3. Compile the Servlet Class:

o Compile the Java Servlet file using the javac command to generate a .class

file.

4. Deploy the Servlet:

o Place the compiled servlet class file in the appropriate directory of the web

application (typically in the /WEB-INF/classes folder).

o Configure the servlet in the web.xml file of the web application or use

annotations.

5. Access the Servlet via Browser:

o Start the Apache Tomcat or another servlet container.

o Access the servlet by entering the appropriate URL (e.g.,

http://localhost:8080/your-webapp/hello).

Code:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;

@WebServlet("/hello")
public class HelloServlet extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<h1>Hello, World!</h1>");
    }
}
```

Output:

RESULT:

The browser will display the message "Hello, World!" returned by the Servlet.