

# HW1 – Due by 11:59pm on Wed, January 29, 2020

---

## Part 1. Reading Assignment

<http://pdf.coreservlets.com/HTTP-Request-Headers.pdf>  
<http://pdf.coreservlets.com/HTTP-Status-Codes.pdf>  
<http://pdf.coreservlets.com/HTTP-Response-Headers.pdf>

## Part 2. Programming Assignment

Write a servlet that displays request headers using:

Enumeration `getHeaderNames()`

Enumeration `getHeaders(String name)`

<https://docs.oracle.com/javase/7/api/javax/servlet/http/HttpServletRequest.html>

## Part 3. Programming Assignment

Create a servlet to read and display user input from the following form (your form design does not need to be the same as the form below. As long as you provide the same fields, that is fine. You may ignore the layout, but if you know Bootstrap and CSS, you may use it.). Use `request.getParameter("fieldname")`.

This is my form

Email:

Password:

Confirm Password:

Upload Your Picture:

Gender: ☐ Male ☐ Female

Select Country:

Hobby: ☐ Cricket ☐ Football

Address

You may ignore the layout and form design.  
Create the form in any way you like.  
Use of Bootstrap and CSS is not required.

## PART 4. Programming Assignment

Redo Part 3 by using `getParameterNames()` method.

<https://docs.oracle.com/javase/7/api/javax/servlet/ServletRequest.html#getParameterNames-->

## PART 5. Programming Assignment

Redo Part 3 by using `getParameterMap()` method.

<https://docs.oracle.com/javase/7/api/javax/servlet/ServletRequest.html#getParameterMap-->

## PART 6. Programming Assignment

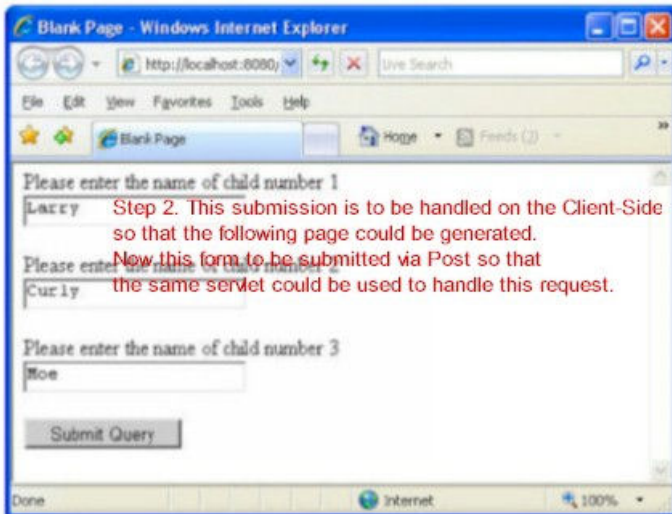
Use only 1 servlet to implement the application whose screenshots are shown in the attached document. Initially form is displayed in the doGet method, and initially form submission will be handled in JavaScript that creates the page as shown in the second screenshot. Now, this form (2nd screenshot) is to be submitted via POST so that the same servlet could handle this request in the same servlet. If you don't know JavaScript, then ignore the first form and just use a static form with 3 fields. In other words, the 2<sup>nd</sup> form below does not need to be created dynamically by JavaScript.

Step 1. Create this form in doGet



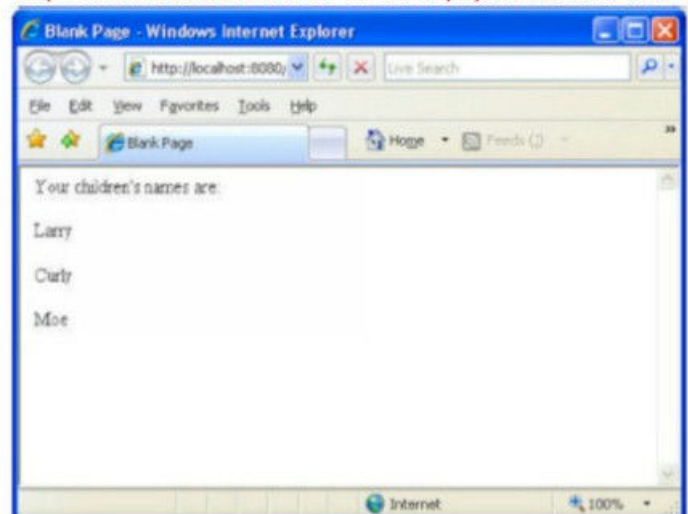
A screenshot of a Windows Internet Explorer browser window. The address bar shows 'http://localhost:8080/'. The page title is 'Blank Page'. The form contains a single text input field with the placeholder text 'How many children do you have?' and a 'Submit Query' button below it.

Step 3. Now handle the form in doPost and display the submitted data.



A screenshot of a Windows Internet Explorer browser window. The address bar shows 'http://localhost:8080/'. The page title is 'Blank Page'. The form contains three text input fields with the following labels: 'Please enter the name of child number 1', 'Please enter the name of child number 2', and 'Please enter the name of child number 3'. The input fields contain the text 'Larry', 'Curly', and 'Moe' respectively. A 'Submit Query' button is at the bottom.

Step 2. This submission is to be handled on the Client-Side so that the following page could be generated.  
Now this form to be submitted via Post so that the same servlet could be used to handle this request.



A screenshot of a Windows Internet Explorer browser window. The address bar shows 'http://localhost:8080/'. The page title is 'Blank Page'. The page displays the text 'Your children's names are:' followed by a list of names: 'Larry', 'Curly', and 'Moe'.