CPRG352 - Web Application Programming

Fall 2021

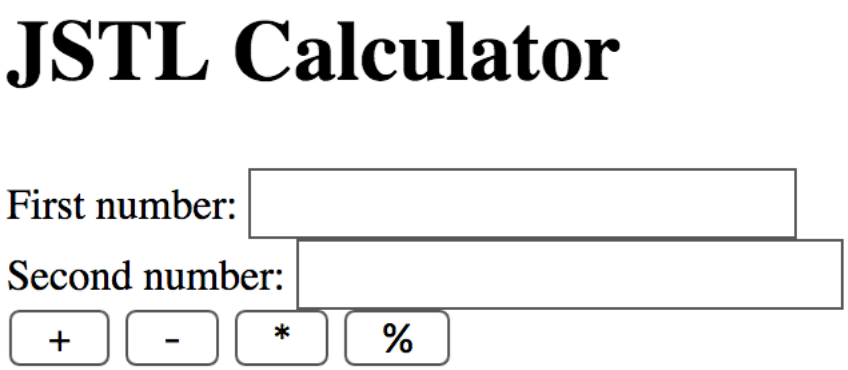
Topic: JSTL usage in JSPs

NOTE: For this lab there should be ***no Java scriptlet code*** in any of the page (i.e. no <%%> tags)

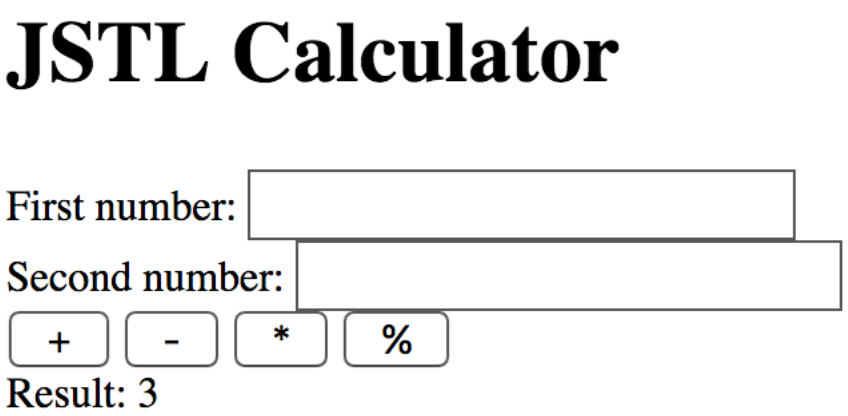
Problem 1: Create a JSTL-based calculator

Create a project called **JSTLCalculator**. This project should implement a default page called **jstlCalculator.jsp** which uses only JSTL to implement a calculator.

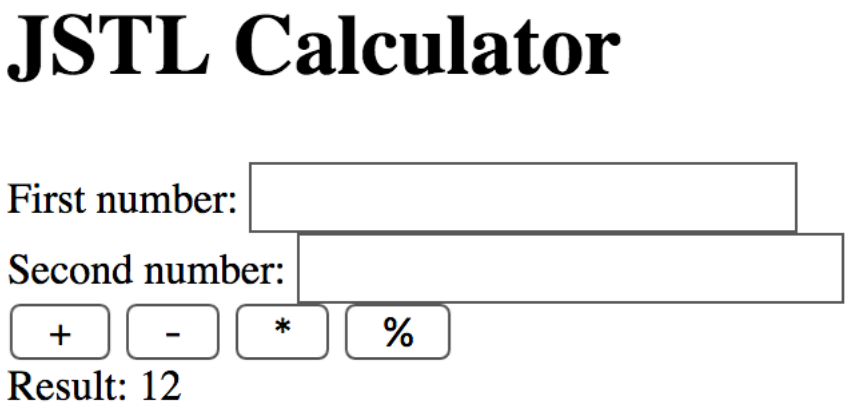
When accessed the page should look like:



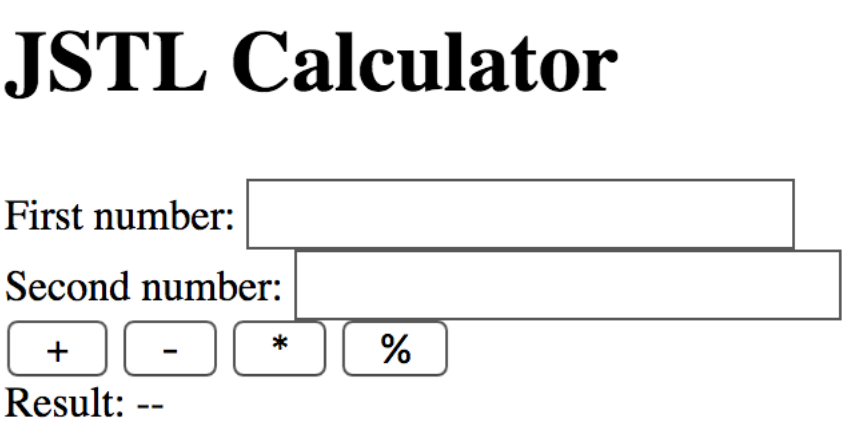
The user can enter two numbers and click on the button for the operation s/he wants to perform, e.g. user enters **1** and **2** and clicks on "+":



If the user enters **3** and **4** and clicks on "*****" then the page should show:



If the user does not submit both values then the page is refreshed with a standard result message, i.e.



Problem 2: Palindrome Finder in JSTL

Create a project called **JSTLpalindromes**. This project will have the user enter a word or sentence and will tell him/her whether or not the entered text is a palindrome.

A palindrome is a piece of text in which the letters and/or numbers are the same when read either forwards or backwards, e.g. **noon**, **abba**, **12321**, etc. When determining if some text is a palindrome we ignore the case of letters, spaces and punctuation symbols, e.g. **abb a**, **N!oon**, **1 2 3 21** are all valid palindromes.

When accessed the page should show:

JSTL Palindromes

Enter text:
Check

If the user enters text and clicks on "Check" then the page should output a message saying if the entered text is a palindrome (based upon the rules above). For example if the user entered "**12noon21**":

JSTL Palindromes

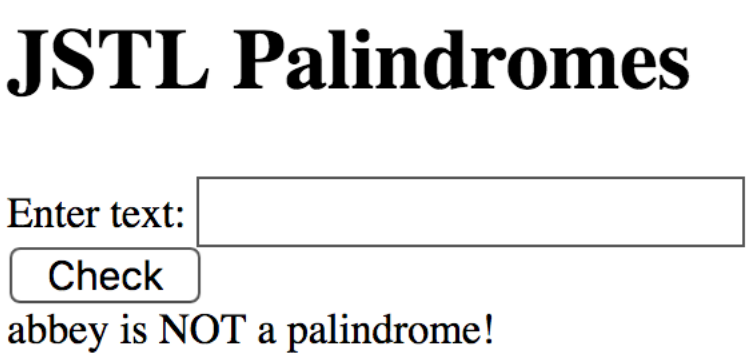
Enter text:
Check
12noon21 is a palindrome

If s/he entered "**N!00n**":

JSTL Palindromes

Enter text:
Check
N!oon is a palindrome

If "**abbey**" is entered:



If nothing at all is entered, then the page can just refresh with no message.

Suggested solution:

- Get the string entered by the user
- Make a separate lower-case version of it
- Loop through the characters in the lower-case version of the string and pull out only the letters and numeric digits in it, concatenate these to a separate string (the "clean string")
- Create a reversed version of the clean string ("reversed clean string")
- If the clean string and the reversed clean string are the same, then the entered text is a palindrome, otherwise it is not

Use the JSTL functions, e.g. **fn:toLowerCase**, **fn:length** and **fn:substring**, to do the string-handling work. Concatenation can be performed using an *EL* expression, e.g. "***${string}${char to concat}***"