

SOEN287 – Winter 2017

Assignment #3

Due Date: By 11:59 PM **March 22th**, 2017

Evaluation: 6% of final mark (see marking rubric at the end of handout)

Late Submission: none accepted

Purpose: The purpose of this assignment is to help you learn the JavaScript and HTML documents, and dynamic documents with JavaScript.

CEAB/CIPS Attributes: This assignment is primarily evaluating your use of JavaScript and HTML, and dynamic documents with JavaScript. (Use of engineering tools).

General Guidelines When Writing Programs:

- Refer to handout for Assignment 1 & 2

Question #1

Write a JavaScript script called "CharacterOccurrences.JS" that inputs several lines of text and a search character and uses String method **indexOf()** to determine the number of occurrences of the character in text.

- A) You have to use the external CSS file called "CharacterOccurrences.CSS" to set the margin of the paragraph to the value 0 (zero).
- B) You have to declare in your HTML form four (04) ids :
 - a. "searchString" as textarea id in paragraph with 4 rows and 55 columns
 - b. "characters" as input id in paragraph with text type and size equal 5
 - c. "searchButton" as input id in paragraph with button type and its value equal "Search"
 - d. "output" as id in paragraph is for the final result.
- C) The JavaScript file (CharacterOccurrences.js) contains three (03) global variables and two (02) functions:
 - a. Global variables :
 - i. searchStr to get the id of "searchString"

- ii. ch to get the id of "characters"
 - iii. outResult to get the id of "output"
- b. The function **getAllDomElement()** that
 - i. Accesses the "searchButton" element and adds the search button using its id by using the existing the function addEventListener(), which takes three (03) arguments: (a) the name of event as a string literal (here is "click"), (b) the function searchOccurrences, and (c) the Boolean value false.
 - ii. Gets all id elements of "searchString", "characters", "output" using the existing function getElementById()
- c. The function searchOccurrences() to search the character we look for and count the number of occurrences of that character.
 - i. 4 local variables: count, chValue, searchStr, result.
 - ii. Use the functions: charAt(0), toLowerCase() and indexOf().
 - iii. If the variable **count** equal 0 (zero) display the message: the character **ch** not found. Otherwise display the result.
- D) At the end of the JavaScript file, finish with this line to fire the load event when a resource and its dependent resources have finished loading:

```
window.addEventListener( "load", getAllDomElement, false );
```

Here is a sample output to illustrate the expected behavior of your program:

File Edit View History Bookmarks Tools Help

Enter some text:

A professor Taleb teaches three courses in this winter term.

Enter a character to search for:

t Search

Results:

6 occurrence(s) of 't' found

File Edit View History Bookmarks Tools Help

Enter some text:

t7iw57iw5u6w7

Enter a character to search for:

7 Search

Results:

3 occurrence(s) of '7' found

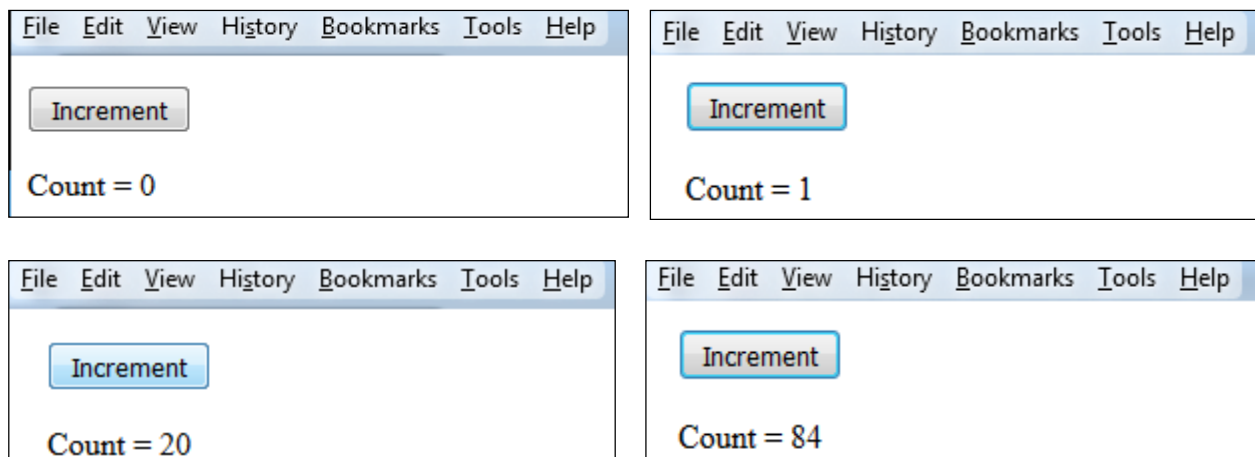
Question #2

Write a script that contains a button (in the external file called "Counter.js") and a counter in a <div> (in the html document called "Counter.html"). The button's event handler should increment the counter each time it's clicked.

- A) You have to use the external CSS file called "Counter.CSS" to set the margin of the paragraph to the desired value, for example 4 or 5, etc.
- B) The JavaScript file (Counter.js) contains two (02) variables and two (02) functions:
1. Global variables: counter = 0 and dom.
 2. The function **getElementBtnIncrement()** that reads the counter element using the existing function getElementById() to access to the element and adds the increment button using its id by using the function addEventListener() to register the event handler, which takes three (03) arguments : a) the name of event as a string literal (here is "click"), (b) the handler function increment, and (c) the Boolean value false.
 3. The function **increment()** that increments the counter by 1 before and assign this new counter value to the dom variable which references the innerHtml.
- C) At the end of the JavaScript file, finish with this line to fire the load event when a resource and its dependent resources have finished loading:

```
window.addEventListener( "load", getElementBtnIncrement, false );
```

Here is a sample output to illustrate the expected behavior of your program:



Question #3

You have a server-side script called "Ampersands.js" that cannot handle any ampersands (&) in the form data. Write a script to convert all ampersands to " and " in the form field when the field loses **focus** element (**blur**).

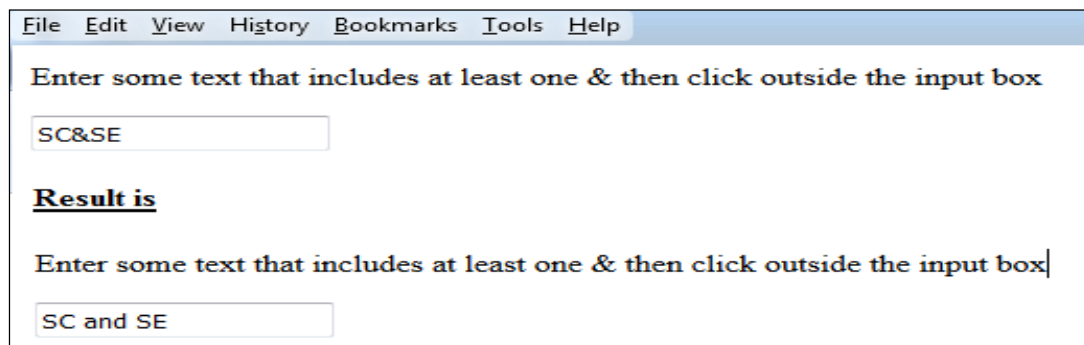
A) Your script contains one (01) variable and two (02) functions:

1. Global variable: dom.
2. The function **getElementAmpersand()** that reads the field element by using the existing function getElementById() to access to the element using its id ("field") specified in your HTML file by using the function addEventListener() to register the event handler, which takes three (03) arguments : (a) the name of event as a string literal (here use "**blur**"), (b) the handler function convertAmpersands, and (c) the Boolean value false.
3. The function "**convertAmpersands()**" that converts all ampersands in the form field to " and " when the field loses focus ("**blur**").
4. The html document called "Ampersand.html", which invokes the "Ampersand.js" in the head section.

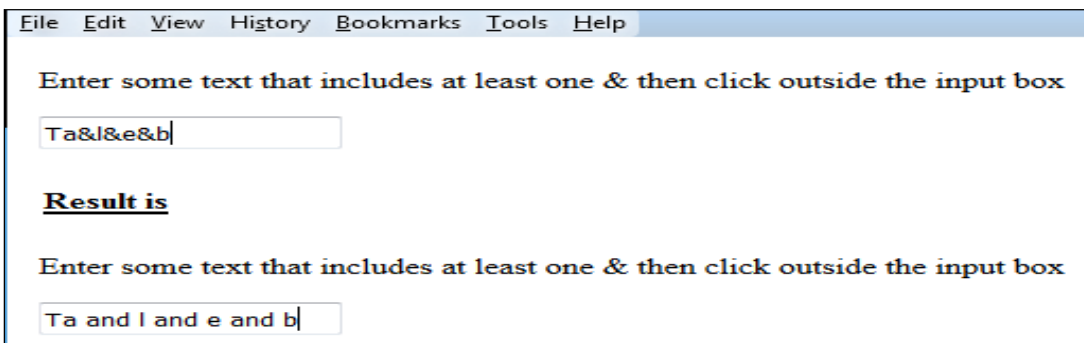
B) At the end of the JavaScript file, finish with this line to fire the load event when a resource and its dependent resources have finished loading:

```
window.addEventListener( "load", getElementAmpersand, false );
```

Here is a sample output to illustrate the expected behavior of your program:



The screenshot shows a web browser window with a menu bar (File, Edit, View, History, Bookmarks, Tools, Help). The page content includes the instruction "Enter some text that includes at least one & then click outside the input box". Below this is an input field containing "SC&SE". Underneath the input field is the heading "Result is". Below the heading is another instruction "Enter some text that includes at least one & then click outside the input box|". At the bottom is an input field containing "SC and SE", demonstrating the successful conversion of the ampersand.



The screenshot shows a web browser window with a menu bar (File, Edit, View, History, Bookmarks, Tools, Help). The page content includes the instruction "Enter some text that includes at least one & then click outside the input box". Below this is an input field containing "Ta&l&e&b|". Underneath the input field is the heading "Result is". Below the heading is another instruction "Enter some text that includes at least one & then click outside the input box". At the bottom is an input field containing "Ta and l and e and b|", demonstrating the successful conversion of multiple ampersands.

Question #4

Copy and paste the following HTML code into a new document:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Exercise 4</title>
</head>
<body>
<div id="container">
<h2>Order Books Online</h2>
<form action="" method="post" id="frm">
<fieldset>
<table border="0">
<tr>
<th>Book</th>
<th>Quantity</th>
<th>Price</th>
</tr>
<tr>
<td>Basic XHTML</td>
<td><input type="text" size="3" id="book_1" /></td>
<td>$19.99</td>
</tr>
<tr>
<td>Intro to PHP</td>
<td><input type="text" size="3" id="book_2" /></td>
<td>$86.00</td>
</tr>
<tr>
<td>Advanced JQuery</td>
<td><input type="text" size="3" id="book_3" /></td>
<td>$55.00</td>
</tr>
</table>
<br /><br />
<input type="submit" value="Place Order" id="sub" />
</fieldset>
</form>
</div>
</body>
</html>
```

Write JavaScript code that is executed during form submission to calculate the cost of each book (based on quantity specified) and the overall total cost. The results should be displayed on the same page beneath the form submission button. Use any CSS at your discretion if desired.

All JavaScript code must be external. If any fields are left blank or do not contain a number, an alert box should display an appropriate error message upon form submission.

The resulted page looks like the following.

Order Books Online

Book	Quantity	Price
Basic XHTML	<input type="text" value="1"/>	\$19.99
Intro to PHP	<input type="text" value="2"/>	\$86.00
Advanced JQuery	<input type="text" value="3"/>	\$55.00

Basic XHTML (Quantity = 1): \$19.99

Intro to PHP (Quantity = 2): \$172

Advanced JQuery (Quantity = 3): \$165

Final Total: \$356.99

Question #5

The running problem to build a website of apartment search (To be continued in Assignment 4). We resume the Nancy's Apartment Search Web site in Assignment 1.

Please create another section called "Expert suggestions" after the section of "What you are looking for". Style it properly and make it invisible at first. Once the user's selections meet the following criteria, use JavaScript to insert the suggested text into this section and make it visible.

- If the size is "More than 51/2" and the location is "Downtown", show the expert suggestion after "Downtown" is selected: "It is very difficult to find an apartment larger than 51/2 in downtown".
- If the size is "4 1/2" and above, the location is "Downtown", and the price range is less than 1000\$, show the expert suggestion after 1000\$ or less options are

selected: "Normally an apartment of 4 ½ and above costs more than 1000\$ in downtown area".

Submitting Assignment #3

- Zip the source code (HTML and JavaScript files only please) of this assignment.
- Naming convention for zip file: Create one zip file, containing all source code files for your assignment using the following naming convention:
 - The assignment is done individually:
The zip file should be called *a#_studentID*, where # is the number of the assignment *studentID* is your student ID number. For example, for the first assignment, student 123456 would submit a zip file named *a3_123456.zip*
- For submission instructions please refer to the course web page.
- Assignments not submitted to the correct location or not in the requested format will not be graded.
- Submit only ONE version of an assignment. If more than one version is submitted the last one will be graded and all others will be disregarded.

Evaluation Criteria of Assignment #3 (25 points)

HTML Document with JavaScript source code	
Question #1 (5 pts.)	
CharacterOccurrences.CSS	0.5 pt.
HTML form four (04) ids	0.5 pt.
Global variables	0.5 pt.
Function getAllDomElement()	1.5 pts.
Function searchOccurrences()	1.5 pts.
Display the result	0.5 pt.

Question #2 (5 pts.)	
Counter.CSS	0.5 pt.
HTML form two (02) ids	0.5 pt.
Global variables	0.5 pt.
Function getElementBtnIncrement()	1.5 pts.
Function increment()	1.5 pts.
Display the result	0.5 pt.
Question #3 (5 pts.)	
HTML form one (01) id	0.5 pt.
Global variables	0.5 pt.
Function getElementAmpersand()	1.5 pts.
Function convertAmpersands()	2 pts.
Display the result	0.5 pt.
Question #4 (5 pts.)	0.5 pt.
Register the event handler	1 pt.
Calculate the sum	1pts.
Display the result	2 pts.
Validation	1 pt.
Question #5 (5 pts.)	
Register the event handler 1	1 pt.
Register the event handler 2	1pts.
Display the result	3 pts.