Concordia University

Department of Computer Science and Software Engineering

SOEN287: Web Programming

Summer 2017

Programming assignment #2

Deadline:	**Monday, July, 31, 2017 @ 23:59**
Late submission:	**Not accepted**
Type of submission	Electronic submission using EAS
Evaluation:	**10% of final mark**

Important note: This is an individual or team assignment and must be submitted electronically using EAS submission system. Teams can be formed by maximum two (2) students.

Content: JavaScript tying into HTML and CSS
Objectives: To obtain a solid understanding of client-side scripting and how it can be used to create dynamic content within a webpage.
Coverage: Client side programming with JavaScript, access Web page elements (DOM elements and CSS attributes), and change their content or values.

Exercise 1

The following set of short questions will focus on getting you familiar with JavaScript functions.

All your JavaScript functions must be declared in the document `head` section and each functions name must be as specified below. To demonstrate the functionality of each method, you must make function calls in the document `body`. Include a heading (`h1… h6`) that indicates which function is being tested before each function demonstration. The use of Global Variables is forbidden!

A. **Function:** `addNumbers`
Parameter(s): Array of numbers
Each element in the array must be added and the summation (answer) must be returned.

B. **Function:** `getCurrentDate`
Parameter(s): None
Retrieve the current date in the format similar to: *Monday, May 10, 2010* and return it.

C. **Function:** `arrayToString`
Parameter(s): Array of words
All the elements of the array must be concatenated into a single string and returned.

D. **Function:** `findMaxNumber`
Parameter(s): Use an arguments array of numbers
From the arguments array, find the element that is the largest and return it.

E. **Function:** `getDigits`
Parameter(s): A String
Scan the string and find all the digits (0-9), concatenate them into a string in the order that they are found and return the string of numbers.

F. **Function:** `reverseString`
Parameter(s): A String
Reverse the entire string (character by character) and return the resulting string.

Exercise 2

Part 1
Given the following Regular Expressions, explain in detail the pattern accepted by each. Pay attention to all the characters being used. *Your answers need not appear within an HTML page.*

A. `/[a-z]*\s[0-9]*/`
B. `/\d+\.\d*/`
C. `/^\d{3}\-\d{4}$/`
D. `/^[A-Z]\d[A-Z]\-\d[A-Z]\d/`
E. `/(July|August)\s+\d{1,2}(\s|,)\d{4}/g`

Part 2
Given the following descriptions, write a possible regular expression that expresses each of them. *Your answers need not appear within an HTML page.*

A. **Matching a Name -** First name must begin with a capital letter and be at least 2 characters long followed by either a (,) or a whitespace. The Last Name should begin with a capital letter and be at least 2 characters long but no greater than 20 characters long.

B. **Matching an Email –** First section should match word characters of any length except zero and must be followed by a @ symbol. The last section should match word characters of any length except zero and must not include an underscore (_). A single period (.) must follow along with the word *com* at the end. No spacing is allowed within the pattern. The pattern need not match exactly (pattern can be a substring of a larger string).

Exercise 3

Given the following HTML page:

```
<html>
     <head>
          <title>SOEN 287 Exercise </title>
     </head>
     <body>
        <p><strong>User Information:</strong></p>
           <div id=”content”></div>
     </body>
</html>
```

Create an embedded JavaScript function named `getUserInfo` that prompts the user with the following two questions after the HTML page has loaded:

- What is your full name?
- How old are you?

The function should then build a string in the form of:

Hi, my name is *FULL-NAME* and I’m *XX* years old.

The string must then be inserted into the container `div` with id: *content*.

Exercise 4

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

```
<?xml version="1.0" encoding="utf-8"?>
<!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>Basics of C++</td>
      <td><input type="text" size="3" id="book_1" /></td>
      <td>$19.99</td>
     </tr>
     <tr>
        <td>Program Development in Perl</td>
        <td><input type="text" size="3" id="book_2" /></td>
        <td>$86.00</td>
     </tr>
     <tr>
        <td>Advanced JavaScript</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.

Exercise 5

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

```
<?xml version="1.0" encoding="utf-8"?>
<!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> SOEN 287 NEW Exercise </title>
<style type="text/css">
fieldset {border:0px;}
#courselist {width:300px;}
#courselist div {border: 1px black solid;padding:10px;}
</style>
</head>
<body>
 <div id="container">
  <h2>Search a Course</h2>
  <form action="" method="post" onsubmit="return searchList()">
   <fieldset>
    Enter the Course Name<br />
    <input type="text" id="search" size="20" /><br />
    <input type="submit" value="Search List" id="sub" />
    <br /><br />
   </fieldset>
  </form>
  <div id="courselist">
   <div id="first">&nbsp;</div>
   <div> Machine Learning </div>
   <div> Image Processing</div>
   <div>Design and Analysis of Algorithms</div>
   <div>Web Programming II </div>
   <div>Advanced JAVA</div>
   <div>Pattern Recognition</div>
  </div>
 </div>
<script type="text/javascript" src="main.js"></script>
</body>
</html>
```

The above XHTML code validates as 1.0 Strict and must not be modified in anyway.

In an external JavaScript file write code that, upon form submission, scans the list of courses for a match (case should be insensitive). If a match is found, change the `div`'s background color to yellow. If no match is found, add the course name to the bottom of the course list (Hint: Read/Write into DOM and Nodes). If the search field is left blank, return focus to the search field and display an alert box with an appropriate message. New searches must clear any yellow background before anything.

Exercise 6: Project

The last exercise of each assignment will be continuous from assignment to assignment and will have the goal of building an e-store website by the end of the course.

Date & Time
In the header area of your website, use JavaScript to display the current date and time. The time should automatically refresh every second.

Items on Sell

In the *inside content* area (see assignment 1) of your sites home page, you should create a small list of items that your store will be selling (your website can be a seller of whatever products you'd like). Each item should have a small thumbnail picture, an associated price, short description and a form button labeled "Add to Cart". The button need not do anything when clicked at the moment.

Registration / Sign-up

In the *side menu* area (see assignment 1) add a link labeled "Register". When clicked, the link should direct the user to a new page: register.html. This new page should include all the elements of your home page. In the *inside content* area, create a form with the following fields:

Text Fields: First name, Last name, Email Address, Phone Number
Password Fields: Password, Confirm Password

Use JavaScript to create some basic client-side validation. When the forms submit button is clicked, you should check to make sure that none of the fields are left blank. You must also check to see if both passwords match entirely. If validation fails, use an alert box to display an appropriate message that informs the user on what needs to be corrected. If validation is successful, do nothing for the time being.

Submission Criteria

Your submission must include a README.txt file that includes the following information:

- Students Full Name
- Students ID
- Difficulties (outlining difficulties will help the TA's focus on repairing weaknesses)
- Additional comments (anything deemed important for marking purposes)

Please give meaningful names to each html file, folder to make the feedback process easier. All files relating to each Exercise must be placed in its own folder named EX-# where # is the number corresponding to the exercise. All the exercise folders must then be placed into a single assignment folder named: *studentIDs*-assignment2

The assignment folder must be compressed into a zip file and submitted through EAS on the correct folder (Programming Assignment#2). For example, for the first assignment, student 123456 would submit a zip file containing all the files and folders related to this assignment named a2_123456.zip, and will upload it to EAS.

Please note that the markers reserve the right to deduct marks if the submission format is not followed as specified. Late submissions are not accepted; penalty for late submission will be 100% (Assignments submitted after the due date will receive a

mark of 0). Also, email submission of assignments will not be accepted under any condition.

Academic integrity:

Students are encouraged to study and work in groups and discuss and share their knowledge with each other. However, copying is strictly prohibited and all assignments to be copies would not receive any marks. Also, those students who are found copying will face severe consequences. Students should be aware and observe the academic integrity & the university's code of conduct. For more information please refer to the course outline.

Evaluation Criteria: Marking schema of assignment no.2

Exercise		Description	Points	
Ex. 1		2 points/each function	12	
Ex.2	Part 1	2 points/each expression	10	14
	Part 2	2 points/each expression	4	
Ex.3		Prompt questions (3 points) string display (3 points)	6	
Ex.4		Display the cost of books (3 points) Display alert box (3	6	

		points)	
Ex.5		Three cases: match is found, no match is found and the search field is left blank. (2 points/each case)	6
Ex.6	Date&Time		2
	Items on sale	Four elements: a picture, a price, a description and a form button (1 point/each element)	4
	Registration	Form display (2 points); non-blank validation(2 points); password validation (2 points)	6
Total points			56

Wishing you success
July 21, 2017