Concordia University

Department of Computer Science and Software Engineering

SOEN287: Web Programming

Winter 2018

Programming Assignment #3

Deadline:	**April 10, 2018 @ 23:59**
Late submission:	**Not accepted**
Type of submission	Electronic submission using EAS
Evaluation:	**11% of final mark**

Content: Server side programming with PHP
Objectives: To obtain a solid understanding of server-side scripting and how it can be used to build dynamic content and real-world applications.
CEAB Attributes: This assignment primarily evaluates use of PHP and HTML, and dynamic documents with PHP. (Knowledge-base and Use of engineering tools).

Exercise 1

The following set of short questions will focus on getting you familiar with how to write PHP functions as well as how to make use of the pre-existing PHP functions. All your PHP functions must be declared in the document body 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! Test all your functions with some relevant input parameter values.

A. **Function:** computefactorial
Parameter(s): Number
Given the number, find its factorial and return it. If the parameter is not an integer number or if it is a negative number, return false.

B. **Function:** findMostFrequent
Parameter(s): String Array
Outputs the string (element) that appears the most frequently within the array with its frequency of repetition. Strings should be case insensitive.

C. **Function:** toUppercaseFirst
Parameter(s): String
For each word in the string, capitalize (uppercase) the first letter, and make sure the rest of the letters of all words are lower case, then return the modified string.

D. **Function:** splitCapitalizeSort
Parameter(s): String
Given the string, place each word into an index array and then capitalize (uppercase) the first letter of all those words in the array, then sort the array alphabetically. The return the resulting array. Duplicated words, if any, are kept.

E. **Function:** dayofNextFriday
Parameter(s): None
Return the date of the next Friday in the following format: DD/MM/YYYY.
(if today is Friday return the date of next Friday)

F. **Function:** findUniqueandSort
Parameter(s): Array of integer numbers
Scan the entire array and print out the array excluding any duplicate elements, then sort the array. Example: input Array a = 1,3,2,1,3, 1 would output 1,2,3.

G. **Function:** sortHash1
Parameter(s): Associative Array
Given an associative array using full-names as keys and annual salaries as values; in a table format, print out the name and corresponding salary of each individual in increasing order (from lowest salary to the highest).

H. **Function:** sortHash2
Parameter(s): Associative Array, code
Write a PHP function to accept an associative array (as follows) and a code as input parameters, and then sort the input associative array as follows. Example of the input array:
array("Jack"=>"55",
"Anita"=>"30","Ramesh"=>"40","Sophia"=>"21","Nastran"=>"41","William"=>"39","David"=>"5") in :

a) if code =1: ascending order sort by value
b) if code =2: ascending order sort by Key
c) if code =3: descending order sorting by Value
d) if code =4: descending order sorting by Key
e) For other values of code: Function only prints ("Wrong Code")

I. **Function:** averageTemp
Parameter(s): Associative Array

Write a PHP function to calculate and display average temperature, and four lowest and highest temperatures.
Recorded temperatures are : [78, 60, 62, 68, 71, -17, 52, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, -10, 72, 65, 80,74, 62, 62, 65, 64, 0, 68, 73, 75, 79, 73, 77]

Expected Outputs:

Average Temperature is : Average Value
List of four lowest temperatures : -17, -10, 0, 52
List of four highest temperatures : ~~79, 80, 81, 85~~ **78, 79, 80, 85**

J. **Function:** findatStartorEnd
Parameter(s): word, string

Write a PHP function to test if a given word occurs at the beginning or at the end of a given string. If yes, it returns: True, otherwise it returns: False

Example for: "I love PHP", "PHP" --→ function returns : True

Example for: "I love PHP and C++", "PHP" --→ function returns : False

Exercise 2

Given the following URI (Uniform Resource Identifier):

http://www.site.com/index.php?age=22&name=Obi+Wan+Kenobi&number=5435432156

Answer the following short questions:

1- What is a query string and what purpose does it serve in server-side programming?
2- Is a query string characteristic of a GET or POST request method?
3- Which part of the URL is considered the query string?
4- How many different data parameters are being passed?

Create a single .php page (name it index.php) that, when passed the above query string as part of the URL processes the data using PHP and outputs it onto the screen in a nicely formatted table that resembles (CSS should be used in order to add some style):

Name	*Display Name data from query String*
Age	*Display Age data from query String*
Phone Number	*Display phone number from query String*

The text in *italics* must be replaced with the appropriate values retrieved from the URI query string. If the query string is omitted from the URL, the above table should be replaced with the message: "No query string data found".

Exercise 3

Create a new .php page (name it track.php) that uses cookies to track the number of times the page has been viewed. If the visitor is looking at the page for the first time, it should display the following message: *Welcome! You are a new customer here.* If the page has been visited more than once, display: *Hello, this is your ## time here* (where ## is the number of visits count). Furthermore, if a page has been visited more than once, also display the date and time that the page was last visited on.

Submission Criteria

Exercise	Description	Points/36
Ex. 1	2 points/each function (10 functions in total)	20
Ex.2	1 point/each question (4 questions in total) 2 points/each line in the table (3 lines in total)	10
Ex.3	"Welcome! You are a new customer here" (2 points) "Hello, this is your ## time here" (2 points) Display the date and time (2 points)	6

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: *studentID*-assignment3

The assignment folder must be compressed into a zip file and submitted through EAS on the correct folder (Programming Assignment_3). 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.