PHP L3: Hands-on Assignments

Estimated Efforts: 2 PDs

Author: ashish.sahu3@wipro.com

No. of Assignments to be Completed	**4**

Environment Setup Details:

Option 1	Software to be installed and available on your local PC/Laptop PHP 7X or PHP 8 preferred (https://www.php.net/downloads.php) You can raise a software request for the above software by following one of the below options: 1. https://wasp.wipro.com/esd **2.** https://mywipro.wipro.com --> My Requests --> IT Services --> Software Requisition

General Instructions:

a. Create separate project for each of the assignments.

b. Capture screen shots of the output of the program wherever applicable and submit it along with the solutions.

c. Submit the src and pom.xml for each project.

Topic 1: PHP

Sub Topic: Advance of String, Operator, Loop, Array functionality and it's features.

Assignment 1: Array, Loop, Function

a) Write a function that returns true if you can partition an array into one element and the rest, such that this element is equal to the product of all other elements excluding itself. For Examples: canPartition([2, 8, 4, 1]) ➞ true
// 8 = 2 x 4 x 1 canPartition([-1, -10, 1, -2, 20]) ➞ false

b) Create a function that takes in an array and returns an array of the accumulating sum.
For Examples: accumulatingArray([1, 2, 3, 4]) ➞ [1, 3, 6, 10]
// [1, 3, 6, 10] can be written as [1, 1 + 2, 1 + 2 + 3, 1 + 2 + 3 + 4]
accumulatingArray([1, 5, 7]) ➞ [1, 6, 13]

c) Create a function that takes a number as its argument and returns an array of all its factors.
For Examples: factorize(12) ➞ [1, 2, 3, 4, 6, 12] factorize(4) ➞ [1, 2, 4]

Assignment 2: String Management

a) Create a function that takes a string, removes all "special" characters (e.g. !, @, #, $, %, ^, &, \, *, (,)) and returns the new string. The only non-alphanumeric characters allowed are dashes -, underscores _ and spaces.
For Examples:
removeSpecialChar("The quick brown fox!") → "The quick brown fox"
removeSpecialChar("%fd76$fd(-)6GvKlO.") → "fd76fd-6GvKlO"

b) Create a function that takes in a sentence and returns the average length of each word in that sentence. Round your result to two decimal places.
For Examples:
averageWordLength("A B C.") → 1.00
averageWordLength("What a gorgeous day.") → 4.00
averageWordLength("Dude, this is so awesome!") → 3.80

c) Create a function that takes a string and returns the number of alphanumeric characters that occur more than once.
For Examples:
duplicateCount("abcde") → 0
duplicateCount("aabbcde") → 2
duplicateCount("Aa") → 0
// Case sensitive

Assignment 3: Sorting, Conditions, Regex and Math

a) Create a function that returns all pairs of numbers in an array that sum to a target. Sort the pairs in ascending order with respect to the smaller number, then order each pair in this order: [smaller, larger].
For Examples:
allPairs([2, 4, 5, 3], 7) → [[2, 5], [3, 4]]
// 2 + 5 = 7, 3 + 4 = 7 allPairs([4, 5, 1, 3, 6, 8], 9) → [[1, 8], [3, 6], [4, 5]]
// Sorted: 1 < 3 < 4; each pair is ordered [smaller, larger]

b) Write a function that takes a string and calculates the number of letters and digits within it. Return the result as an array.
For Examples:
countAll("Hello World") → array("LETTERS" => 10, "DIGITS" => 0)
countAll("H3ll0 Wor1d") → array("LETTERS" => 7, "DIGITS" => 3)
countAll("149990") → array("LETTERS" => 0, "DIGITS" => 6)

c) Create a function that takes an array of increasing letters and return the missing letter.
For Examples:
missingLetter(["a", "b", "c", "e", "f", "g"]) → "d"
missingLetter(["O", "Q", "R", "S"]) → "P"
missingLetter(["t", "u", "v", "w", "x", "z"]) → "y"

d) Create a function that finds how many prime numbers there are, up to the given integer.
For Examples:
primeNumbers(20) → 8

```
// 2, 3, 5, 7, 11, 13, 17 and 19
primeNumbers(30) ➞ 10
// 2, 3, 5, 7, 11, 13, 17, 19, 23 and 29
```

Assignment 4: PHP / MySQL

a) Create a php page and create a user form which asks for marks in five subjects out of 100 and then displays the mark sheet of the student. The format is as follows:
 Name of Student*:
 Marks in Each Subject
 1) Subject 1* :
 2) Subject 2* :
 3) Subject 3* :
 4) Subject 4* :
 5) Subject 5* :
 Total Marks Obtained:
 Total Marks:
 Percentage:
 Note: All the entries marked (*) are to be input by the user. And use a submit button to post the entries in the form using the POST method. The data gets saved in a MySQL Database named "result" and table named "class1". with each user assigned a unique numeric id (AUTO INCREMENT TYPE).The following is the format to store the data in the result database
 1) id(BIG INT),
 2) name(CHAR),
 3) sub1(INT),
 4) sub2(INT),
 5) sub3(INT),
 6) sub4(INT),
 7) sub5(INT),
 8) total obtained (INT),
 9) total marks(INT)
 10) percent(FLOAT).

b) In the previous assignment use the following data:
 Name: Rohan
 Marks in Subject(s): 55,66,77,88,76
 Now write a PHP Script to update the values in the database with the new marks in subject 5 as "99" and recalculating and updating database entries: total obtained and percent