Laboratory Exercise #9 – Arrays

An array is made by using assigning to the variable a list of elements through the array keyword:

$var = array(element1, element2, …)

A shortcut would be to use brackets:

$var = [element1, element2, …]

1. Write a script which will display the following string: The memory of that scene for me is like a frame of film forever frozen at that moment: the red carpet, the green lawn, the white house, the leaden sky. The new president and his first lady. - Richard M. Nixon
2. The colors 'red', 'green' and 'white' will come from the array variable $color.
3. From the color variable, output in such a way that the output will be:
    a. Red
    b. Green
    c. White
4. For the array:
   array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40") , do the following:
    a. Sort by value, ascending
    b. Sort by key, ascending
    c. Sort by value, descending
    d. Sort by key, descending

   In this exercise, refer to the php.net documentation under array functions

Note: for the above exercises particularly number 2 and 3, this requires the use of loops (you can use foreach, for, while loops)

5. For the following arrays:
   $array1 = array(“apple”,”orange”,”peanut”)
   $array2 = array("w3resource", "com");

   merge the two arrays together.

Laboratory Exercise # 10 – Functions

1. Create a set of functions that will do the following
    a. Add two numbers together
    b. Subtract two numbers together
    c. Multiply two numbers
    d. Divide two numbers

Make sure the the return will be outputed by the following format:
"Add/Subtract number1 and number2 equals result"

2. A factorial is the product of all integers less than or equal to n. For example 5! = 5 x 4 x 3 x 2 x 1 = 120. Make a function that will return a factorial of a given number.

Laboratory Exercise # 11 – Objects and Classes

1. Write a simple PHP class which displays the following string. 'MyClass class has initialized !'. Hint: use the constructor method to output the string.
2. Write a simple PHP class which displays an introductory message like "Hello All, I am Scott", where "Scott" is an argument value of the method within the class.
3. From the functions you created in Exercise 10, create a class named Mathematics with the following functions as your class methods. And make an output example of each method by invoking the class.
4. Make a class named Mathematics2 which extends the Mathematics class and adds a method named author which simply echos out your name.

Laboratory Exercise #12 – Forms and working with files

In this exercise we use the include or include_once keyword. This includes a file you made so that it will be used in your php file.

1. Using your class Mathematics (saved as math.php), create a page named addresult.php and add.php where add.php is your form (with two inputs), and addresult.php is where you use your math.php class to call the add method to output the result of two added numbers.

   The example goes like this:

   add.php:

```
<html>
        <body>
                <form method=post action="addresult.php">
                        <input type=text name=num1 /> <br />
                        <input type=text name=num2 /> <br />
                        <input type=submit value="Add" name=submit />
                </form>
        </body>
</html>
```

   addresult.php (hint):

```
<?php
include "math.php"; // Can be include_once or require_once
```

```
$math = new Mathematics();
…
?>
```

2. Create a file named name.txt and place it in your htdocs folder.
3. Create a program that will print out what is in the name.txt file.
4. From #3, output the result to a new file named name2.txt.