INFO 151 – Web Systems and Services
Week #8: Coursework
PHP Reading and Practical Exercises

1. Read and understand the basics of PHP and server-side server systems. Read **ONLY** from the recommended course textbook (**NOT** any other books which may provide incorrect information)
2. For PHP the required reading is:
    a. Chapter 12: PHP Fundamentals (variables, strings, an arrays) – pages 299-325
    b. Chapter 13: PHP Fundamentals (functions, objects, and flow control) – pages 327-365
    c. Chapter 14: (Working with cookies and user sessions) – pages 367-383
    d. Chapter 15: (Working with web-based forms) – pages 385-423
3. For Database, SQL, and MySQL the required reading is:
    a. Chapter 16: (Understanding the database design process) – pages 425-436
    b. Chapter 17: (Learning basic SQL commands) – pages 437-485
    c. Chapter 18: (Interacting with MySQL using PHP) – pages 487-499
4. Exercises (for HTML, JavaScript, Database, PHP, and MySQL):
    a. At the end of each chapter in the course textbook you will find (a) a Question and Answer (Q&A) section and (b) Workshop section with questions and exercises.
    b. For all the chapters we have covered in this course: work through the Q&A sections and the Workshop exercises and questions (this will be important for the final exam).
5. Read and understand the similarities and differences between JavaScript and PHP syntax
6. Read and understand the nature of PHP arrays and how they differ from JavaScript arrays
7. Create a new PHP project in the NetBeans IDE and complete the set practical programming examples shown below

The coursework exercises to be completed in the laboratory session and as coursework
For the Weeks #4, #5, #6 you were required to complete the JavaScript programming exercises to practice the use of JavaScript. You are now required to revise the JavaScript code and complete the following programming PHP exercises with the output revised to read “PHP” (not) “JavaScript”.

PHP Exercises

As we have seen PHP and JavaScript are very similar. However, there are differences in the syntax as introduced in the tutorials and in the course resources provided.
You will find full details of functions in PHP plus other features (such as scope etc) of the language in Chapter 13 of the recommended course textbook and the tutorial slides (do not use other books as they may not provide the correct information or code examples).

To practice the use of PHP and the differences in the syntax you are required to complete the following exercises as follows:

1. Create a PHP project for each exercise (with a suitable memorable project name)
2. Create the HTML web page (*index.php*) with the correct HTML <tags> and PHP script <tags>

Exercise #1

1. Create the following PHP variables (*operands*):
    a. $n1 = 5;
    b. $n2 = 9;

c. $n3 = 2;
d. $n4 = 4;
e. $n5 = 6;
f. $res;

2. Create a function called: *compute()*
3. Considering operator precedence write a PHP script using the add (+), multiplication (*), divide (/), and subtraction (-) operators
    a. Write a PHP script that uses all of the *operators*
    b. The output (the result) must be 1
    c. Using echo output the result to the web browser
    d. Using printf() output the result as a float with 2 decimal places

Exercise #2:

- Complete the following exercises taken from Chapter 13 of the course textbook):
    - Create a function that accepts four string variables, returns a string that contains an HTML table element, and encloses each of the variables in its own cell;
    - Create a class called baseCalc() that stores two numbers as properties. Next, create a method called calculate() that prints the numbers to the browser;
    - Now create classes called addCalc(), subCalc(), mulCalc(), and divCalc() that inherit functionality from baseCalc() but override the calculate() method, and then print the appropriate totals to the browser.

Exercise #3

- Write a program to:
    - Take in three inputs (diameter, radius, height)
    - Calculate the area of a circle
    - Calculate the circumference of a circle
    - Calculate the volume of a cylinder
    - Output the three results in three correctly formatted strings

Note: In all programs the string(s) (using echo and printf() (formatted to 2 decimal places) must show:

- The formula
- The calculation
- The variable values
- The result

The following screenshots (exercises (#4 to #17) show the output of programs in a web page. The programs have been written in JavaScript:

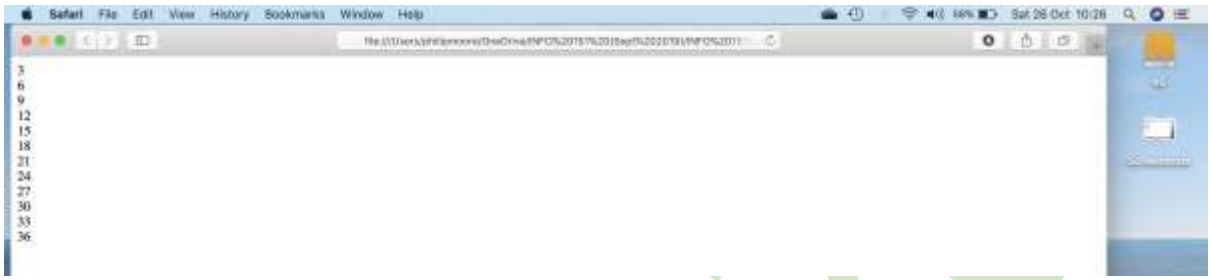
- Create a NetBeans project for each exercise and create a program to reproduce exactly the output shown in the screenshots using both JavaScript and PHP (note the similarities and differences in the syntax).

Exercise #4

Exercise #5

Exercise #6

Exercise #7



Exercise #8

Exercise #9

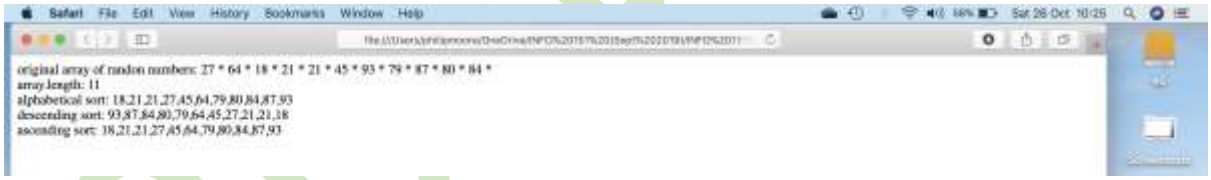
Exercise #10

Exercise #11

Exercise #12

Exercise #13

Exercise #14



Exercise #15

Exercise #16

Exercise #17

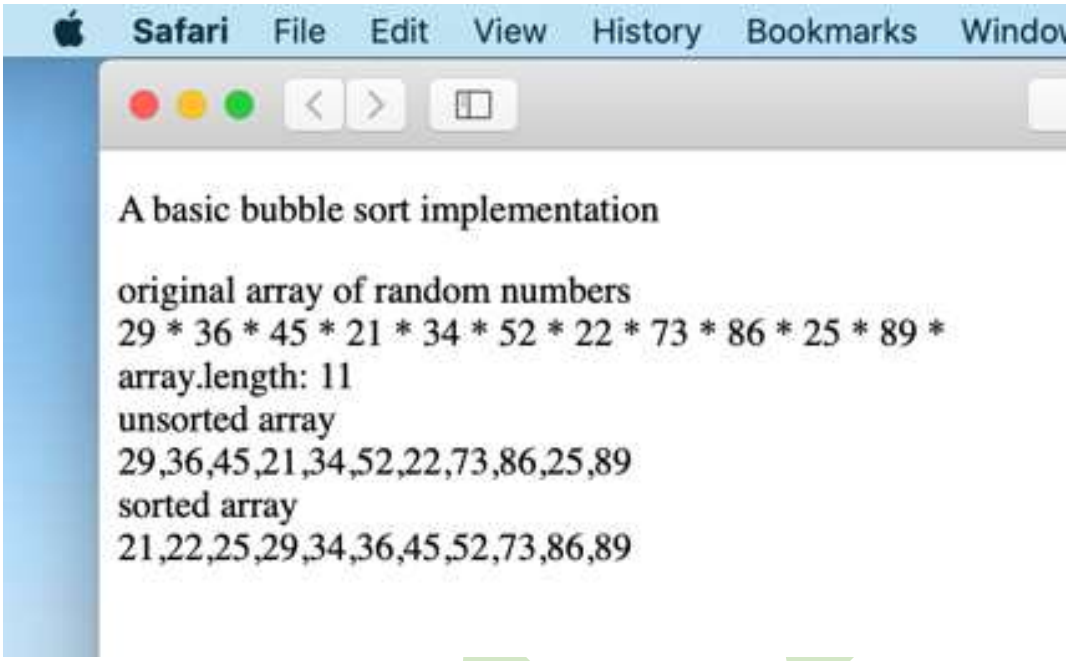
Exercise #18 (10 random numbers)

The following example shows an implementation of a JavaScript *array* in a simple *Bubble Sort* program. The program creates pseudo random numbers and sorts the numbers in ascending order. The program code is shown in the *EMACS* editor. You are required to:

- Revise the JavaScript code as shown using PHP with 10 random numbers

```
<script>
  var arr = [];
  var i;
  document.write("original array of random numbers <br>");
  for(i = 0; i <= 10; i++) {
          //var x = (Math.random() * 100 + 1);
          var x = (Math.floor(Math.random() * 100 + 1));
          arr[i] = x;
          document.write(arr[i] + " *  ");
      }
  document.write("<br>array.length: " + arr.length + "<br>");
  document.write("unsorted array <br>");
  document.write(arr + "<br>");
  var n = arr.length;
  var temp = 0;
  for(var i=0; i < n; i++){
      for(var j=1; j < (n-i); j++){
          if(arr[j-1] > arr[j]){
              //swap elements
              temp = arr[j-1];
              arr[j-1] = arr[j];
              arr[j] = temp;
          }
      }
  }
  document.write("sorted array");
  document.write("<br>" + arr);
</script>
```

- Run the program and correct any errors in the program code until the program runs correctly as shown in the screenshot below.



- Your output should be exactly as shown in the following screenshot. If the program will not run (or) the output is not as shown there are errors in the program code: correct the errors in the program code until the output is correct.