DEPARTMENT OF COMPUTER STUDIES

**(Applications Development and Emerging Technologies)**

**TECHNICAL-SUMMATIVE ASSESSMENT**

**1**

**PHP OUTPUT, VARIABLE FAMILIARIZATION, OPERATORS AND CONTROL STRUCTURE**

**Student Name / Group**

**Name: Patrick Andrew Tria**

**Name** **Role**

**Members (if Group):**

**Section: 1A TN31**

**Professor: Mr. Abraham Magpantay**

1. **PROGRAM OUTCOME/S (PO) ADDRESSED BY THE LABORATORY EXERCISE**
   * Design, implement and evaluate computer-based systems or applications to meet desired needs and requirements.

**II. COURSE LEARNING OUTCOME/S (CLO) ADDRESSED BY THE LABORATORY EXERCISE**

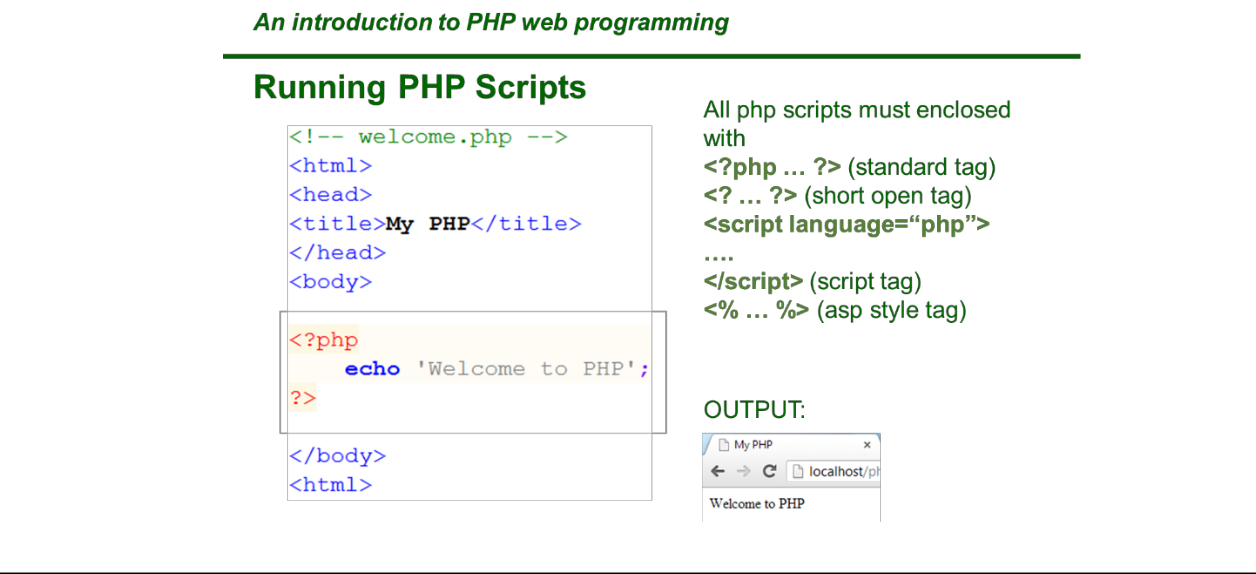
* + Understand and apply best practices and standards in the development of website.

**III. INTENDED LEARNING OUTCOME/S (ILO) OF THE LABORATORY EXERCISE**

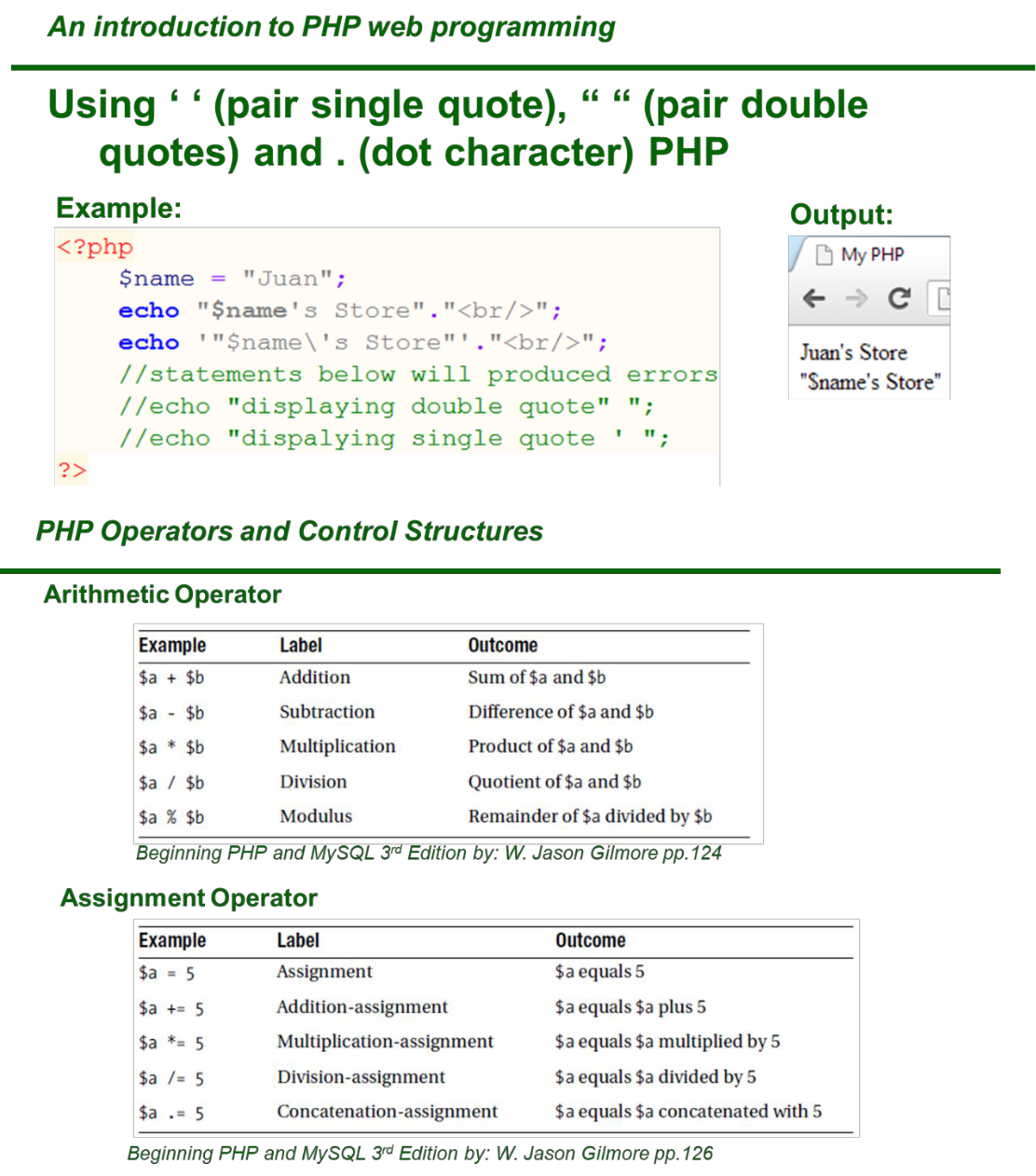
At the end of this exercise, students must be able to:

* + - Familiarize various Web Architecture, tools that used in PHP
    - The basic understanding before using PHP
    - Familiarize in environment of web developing
    - Use of comments, variables and Echo / Print
    - To understand the different types of operators that are available on PHP.
    - To know what is operator precedence and operator associativity in PHP.
    - To use escape sequence properly in the program.
    - To know the different approach of control structures.
    - To know the fundamentals syntax for conditional and looping structures.
    - To properly use the compound expression using the logical operators.
    - To know the rules of break, continue, and goto statements.

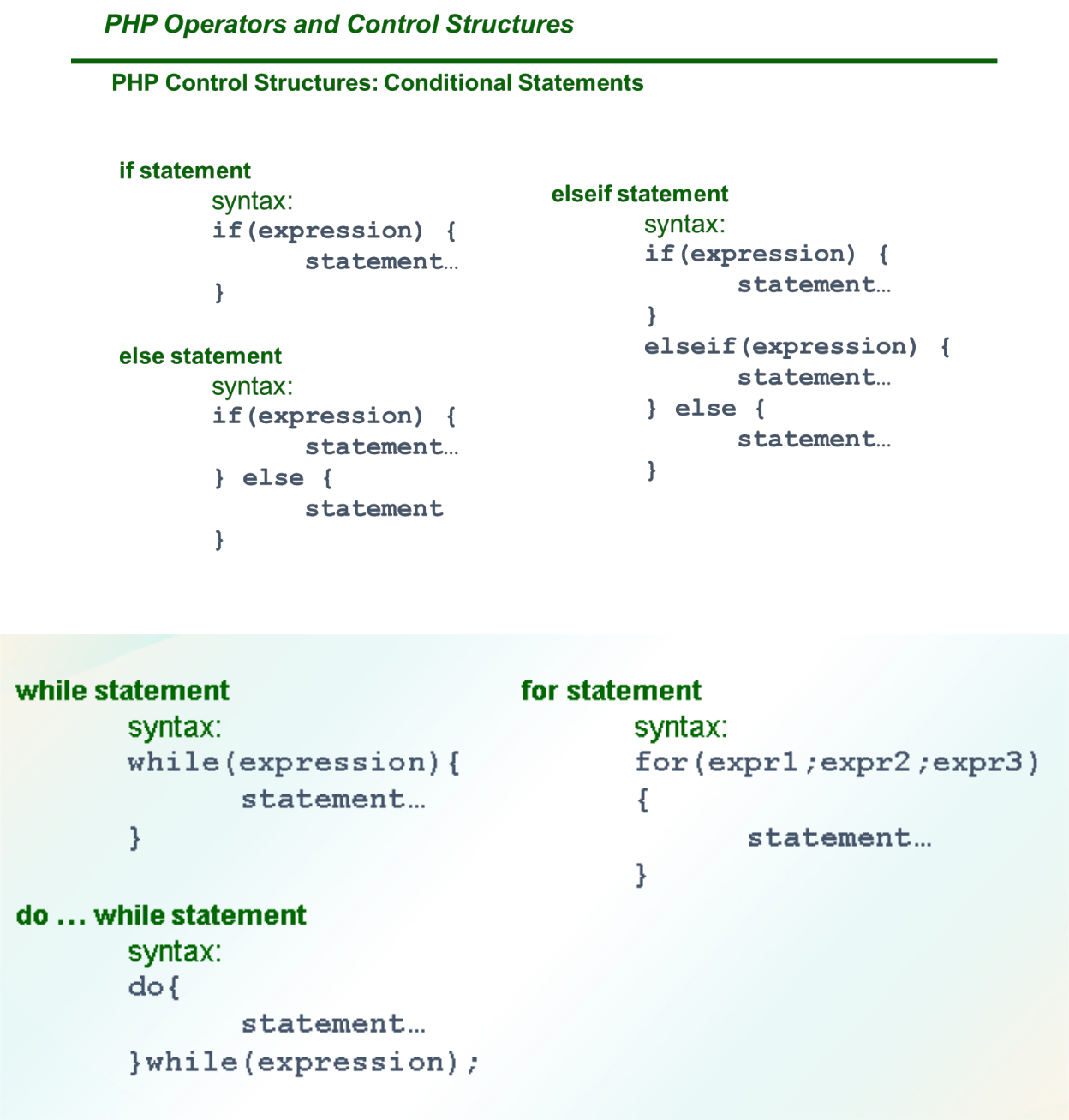
**IV. BACKGROUND INFORMATION**



**Applications Development and Emerging Technologies** **Page 2 of 9**



**Applications Development and Emerging Technologies** **Page 3 of 9**



1. **GRADING SYSTEM / RUBRIC (please see separate sheet)**

**VI. LABORATORY ACTIVITY**

* 1. **The student will create a student registration form using HTML and CSS with the integration of PHP Scripts please refer to the attached image for the example.**
  2. **All user entries from the student registration form will be converted in variables**

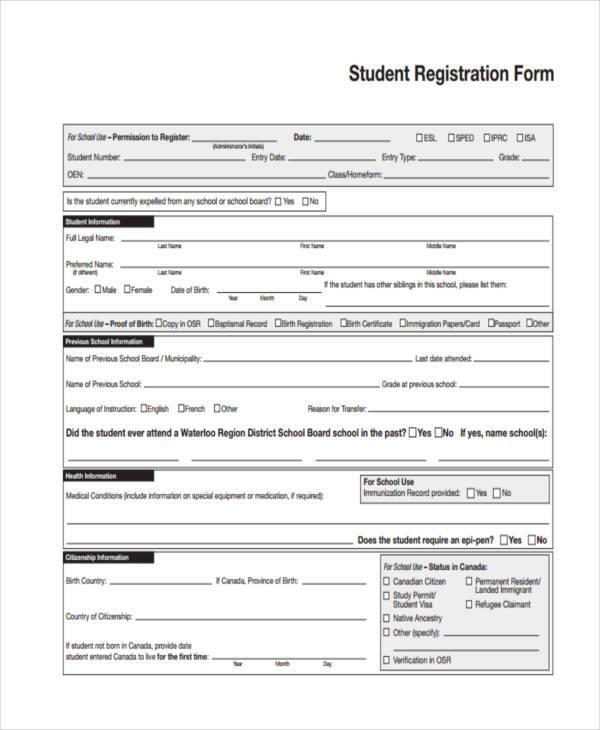


**Applications Development and Emerging Technologies** **Page 4 of 9**

1. **In the output, they need to call for the declared variables and do some string formats like name cases and numbers**

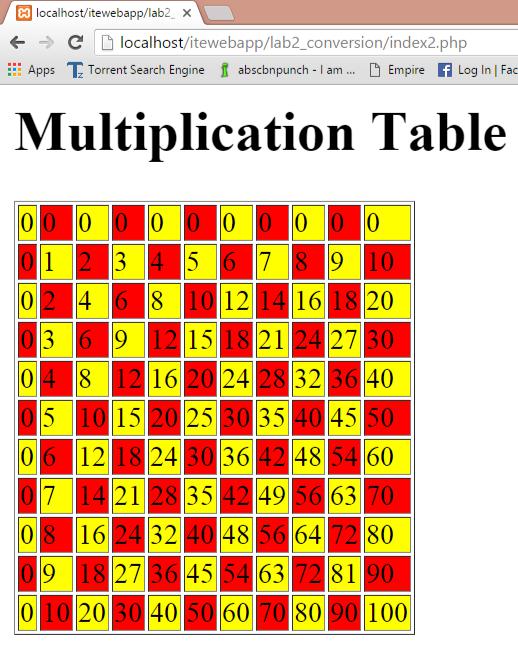


**Applications Development and Emerging Technologies** **Page 5 of 9**



**Applications Development and Emerging Technologies** **Page 6 of 9**

1. Create a php program that will display the same output below. Use control structures to display the multiplication table with alternating color.



1. Using PHP scripts create / simulate a fibonnaci sequence.
2. Using PHP scripts create / simulate a factorial



**n! = n·(n – 1)·(n – 2) · · · 3·2·1**

***Snip and paste your source codes here. Snip it directly from the IDE so that colors of the codes are preserved for readability. Include additional pages if necessary.***



**Applications Development and Emerging Technologies** **Page 7 of 9**

**VII. QUESTION AND ANSWER**

1. **What is a variable?**

Variables are like containers or placeholders for your values that you have to store, edit, manipulate and etc.

1. **What are the rules in creating a variable?**

In PHP, the rules in creating a variable is that it must start with the dollar sign ($) to indicate that it is a variable.

1. **Is it important to know HTML and CSS before using PHP? Explain**

Yes, upon my experiences so far with coding PHP with no background at all at HTML and CSS, I’ve been struggling thus consuming more time in doing or creating the front end.

1. **What is the difference between operator precedence and operator associativity?**

Operator precedence determines which operator will be performed first inside an expression with more than one operator of different precedence. Meanwhile, Operator associativity is used when two operators of same precedence appear in an expression. It can be either left to right or vice versa.

1. **What are the different control structures? Explain each**

The different control structures are Conditional Statements and Control Loops. Conditional Statements are If, Else, Else if, and Switches. If statements execute the code if a condition is true and other part is false. The else if statements execute different codes that have more than two conditions. Lastly, switch for selecting one of the many blocks of codes to be executed.

The Control Loops are composed of for, foreach, while and do while. For goes through a block of code in specified number of times, foreach loops throughout the elements in an array. The while loop goes through a block of code as long as the condition specified is true. Lastly, the do while executes the block of code inside the do as long as the while condition is satisfied.

1. **Explain the rules of break, continue, and goto statements.**

The break is used to jump out of the loop when a condition inside the loop is satisfied. Meanwhile, the continue breaks one iteration in the loop depending if the condition is satisfied and will continue the next iteration of the loop unlike break. Goto statements allows you to jump inside the block of codes starting from where you place the goto statements in the code.

**VIII. REFERENCES**

1. <https://www.w3schools.com/php/func_string_echo.asp>
2. <https://www.w3schools.com/css/>
3. <https://www.w3schools.com/html/>
4. <https://www.w3schools.com/php/php_variables.asp>
5. <https://www.w3resource.com/php/operators/arithmetic-operators.php>
6. <https://www.tutorialspoint.com/php/php_arithmatic_operators_examples.htm>
7. <https://www.w3schools.com/php/php_if_else.asp>
8. <https://www.w3schools.com/php/php_switch.asp>
9. <https://www.w3schools.com/php/php_looping.asp>
10. <https://www.w3schools.com/php/php_looping_while.asp>
11. <https://www.w3schools.com/php/php_looping_do_while.asp>
12. <https://www.w3schools.com/php/php_looping_for.asp>
13. <https://www.w3schools.com/php/php_looping_foreach.asp>
14. <https://www.w3schools.com/php/php_looping_break.asp>

**Note: The following rubrics/metrics will be used to grade students’ output.**



**Applications Development and Emerging Technologies** **Page 8 of 9**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Program (100** | **(Excellent)** | **(Good)** | **(Fair)** | **(Poor)** |
| **pts.)** |  |  |  |  |
| **Program** | Program executes | Program executes | Program executes | Program does not |
| **execution (20pts)** | correctly with no | with less than 3 | with more than 3 | execute **(10-** |
|  | syntax or runtime | errors **(15-17pts)** | errors **(12-14pts)** | **11pts)** |
|  | errors **(18-20pts)** |  |  |  |
| **Correct output** | Program displays | Output has minor | Output has | Output is incorrect |
| **(20pts)** | correct output | errors **(15-17pts)** | multiple errors | **(10-11pts)** |
|  | with no errors |  | **(12-14pts)** |  |
|  | **(18-20pts)** |  |  |  |
| **Design of output** | Program displays | Program displays | Program does not | Output is poorly |
| **(10pts)** | more than | minimally | display the | designed **(5pts)** |
|  | expected **(10pts)** | expected output | required output |  |
|  |  | **(8-9pts)** | (**6-7pts)** |  |
| **Design of logic** | Program is | Program has | Program has | Program is |
| **(20pts)** | logically well | slight logic errors | significant logic | incorrect **(10-** |
|  | designed **(18-** | that do no | errors **(3-5pts)** | **11pts)** |
|  | **20pts)** | significantly |  |  |
|  |  | affect the results |  |  |
|  |  | **(15-17pts)** |  |  |
| **Standards** | Program code is | Few inappropriate | Several | Program is poorly |
| **(20pts)** | stylistically well | design choices | inappropriate | written **(10-11pts)** |
|  | designed **(18-** | (i.e. poor variable | design choices |  |
|  | **20pts)** | names, improper | (i.e. poor variable |  |
|  |  | indentation) **(15-** | names, improper |  |
|  |  | **17pts)** | indentation) **(12-** |  |
|  |  |  | **14pts)** |  |
| **Delivery** | The program was | The program was | The program was | The program was |
| **(10pts)** | delivered on time. | delivered a day | delivered two | delivered more |
|  | **(10pts)** | after the deadline. | days after the | than two days |
|  |  | **(8-9pts)** | deadline. **(6-7pts)** | after the deadline. |
|  |  |  |  | **(5pts)** |



**Applications Development and Emerging Technologies** **Page 9 of 9**