DEPARTMENT OF COMPUTER STUDIES

**(Applications Development and Emerging Technologies)**

**PRE-SUMMATIVE ASSESSMENT**

**4**

**PHP PRE DEFINED FUNCTIONS**

**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**

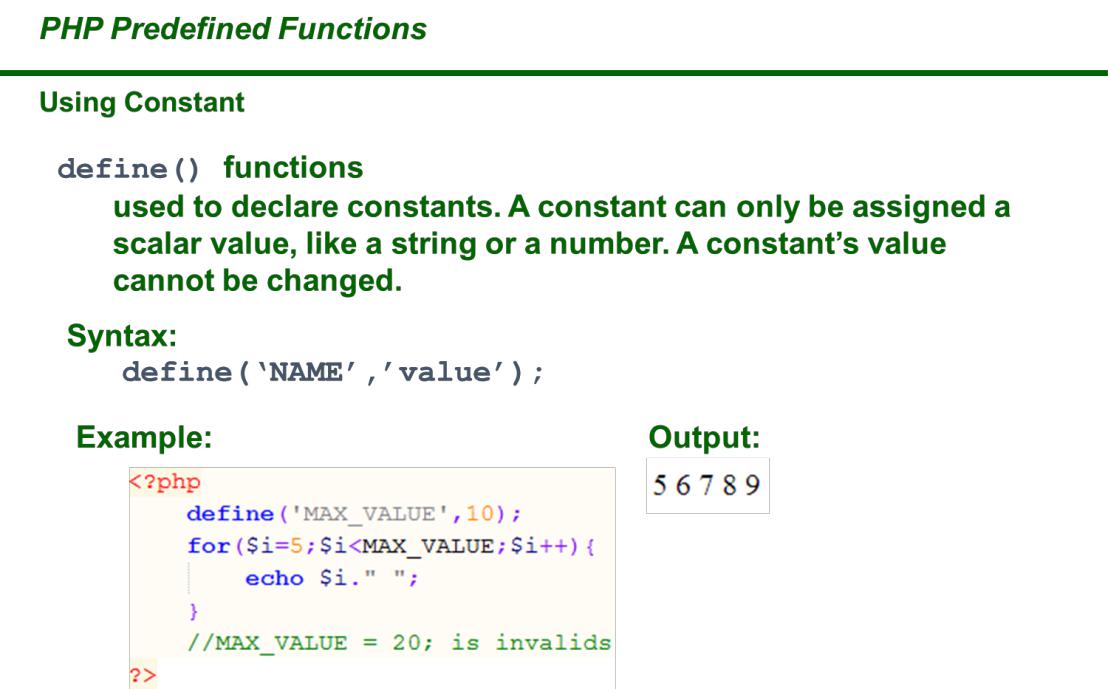
* + Understand and apply best practices and standards in the development of website.

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

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

* + - To know how to include separate PHP code in the main page for code enhancement.
    - To be familiar with the use of common predefined function such as define, include, and require.
    - To use different available mathematical function for manipulating numbers.

**IV. BACKGROUND INFORMATION**



**PHP Constants**

Constants are like variables except that once they are defined they cannot be changed or undefined.

A constant is an identifier (name) for a simple value. The value cannot be changed during the script.



**Applications Development and Emerging Technologies** **Page 2 of 11**

A valid constant name starts with a letter or underscore (no $ sign before the constant name).

**Note:** Unlike variables, constants are automatically global across the entirescript.

Create a PHP Constant

To create a constant, use the define() function.

Syntax

define(*name*, *value*, *case-insensitive*)

Parameters:

* *name*: Specifies the name of the constant
* *value*: Specifies the value of the constant
* *case-insensitive*: Specifies whether the constant name should be case-insensitive. Default is false

Example

Create a constant with a **case-sensitive** name:

<?php

define("GREETING", "Welcome to PHP Programming!"); echo GREETING;

?>

PHP Include Files

The include (or require) statement takes all the text/code/markup that exists in the specified file and copies it into the file that uses the include statement.

Including files is very useful when you want to include the same PHP, HTML, or text on multiple pages of a website.

PHP include and require Statements



**Applications Development and Emerging Technologies** **Page 3 of 11**

It is possible to insert the content of one PHP file into another PHP file (before the server executes it), with the include or require statement.

**The include and require statements are identical, except upon failure:**

* require will produce a fatal error (E\_COMPILE\_ERROR) and stop the script
* include will only produce a warning (E\_WARNING) and the script will continue

So, if you want the execution to go on and show users the output, even if the include file is missing, use the include statement. Otherwise, in case of FrameWork, CMS, or a complex PHP application coding, always use the require statement to include a key file to the flow of execution. This will help avoid compromising your application's security and integrity, just in-case one key file is accidentally missing.

Including files saves a lot of work. This means that you can create a standard header, footer, or menu file for all your web pages. Then, when the header needs to be updated, you can only update the header include file.

Syntax

include '*filename*';

or

require '*filename*';

PHP include Examples

Example 1

Assume we have a standard footer file called "footer.php", that looks like this:

<?php

echo "<p>Copyright &copy; 1999-" . date("Y") . " W3Schools.com</p>"; ?>

To include the footer file in a page, use the include statement:



**Applications Development and Emerging Technologies** **Page 4 of 11**

Example

<html>

<body>

<h1>Welcome to my home page!</h1>

<p>Some text.</p>

<p>Some more text.</p>

<?php include 'footer.php';?>

</body>

</html>

Example 2

Assume we have a standard menu file called "menu.php":

<?php

echo '<a href="/default.asp">Home</a> -

<a href="/html/default.asp">HTML Tutorial</a> - <a href="/css/default.asp">CSS Tutorial</a> -

<a href="/js/default.asp">JavaScript Tutorial</a> - <a href="default.asp">PHP Tutorial</a>'; ?>

All pages in the Web site should use this menu file. Here is how it can be done (we are using a <div> element so that the menu easily can be styled with CSS later):

Example

<html>

<body>

<div class="menu">

<?php include 'menu.php';?>

</div>

<h1>Welcome to my home page!</h1>

<p>Some text.</p>

<p>Some more text.</p>



**Applications Development and Emerging Technologies** **Page 5 of 11**

</body>

</html>

Example 3

Assume we have a file called "vars.php", with some variables defined:

<?php

$color='red';

$car='BMW';

?>

Then, if we include the "vars.php" file, the variables can be used in the calling file:

Example

<html>

<body>

<h1>Welcome to my home page!</h1>

<?php include 'vars.php';

echo "I have a $color $car.";

?>

</body>

</html>

PHP include vs. require

The require statement is also used to include a file into the PHP code.

However, there is one big difference between include and require; when a file is included with the include statement and PHP cannot find it, the script will continue to execute:

Example

<html>

<body>

<h1>Welcome to my home page!</h1>



**Applications Development and Emerging Technologies** **Page 6 of 11**

<?php include 'noFileExists.php';

echo "I have a $color $car.";

?>

</body>

</html>

If we do the same example using the require statement, the echo statement will not be executed because the script execution dies after the require statement returned a fatal error:

Example

<html>

<body>

<h1>Welcome to my home page!</h1>

<?php require 'noFileExists.php';

echo "I have a $color $car.";

?>

</body>

</html>

PHP String Functions

In this chapter we will look at some commonly used functions to manipulate strings.

strlen() - Return the Length of a String

The PHP strlen() function returns the length of a string.

Example

Return the length of the string "Hello world!":

<?php

echo strlen("Hello world!"); // outputs 12 ?>



**Applications Development and Emerging Technologies** **Page 7 of 11**

str\_word\_count() - Count Words in a String

The PHP str\_word\_count() function counts the number of words in a string.

Example

Count the number of word in the string "Hello world!":

<?php

echo str\_word\_count("Hello world!"); // outputs 2 ?>

strrev() - Reverse a String

The PHP strrev() function reverses a string.

Example

Reverse the string "Hello world!":

<?php

echo strrev("Hello world!"); // outputs !dlrow olleH ?>

strpos() - Search For a Text Within a String

The PHP strpos() function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.

Example

Search for the text "world" in the string "Hello world!":

<?php

echo strpos("Hello world!", "world"); // outputs 6 ?>



**Applications Development and Emerging Technologies** **Page 8 of 11**

**Tip:** The first character position in a string is 0 (not 1).

str\_replace() - Replace Text Within a String

The PHP str\_replace() function replaces some characters with some other characters in a string.

Example

Replace the text "world" with "Dolly":

<?php

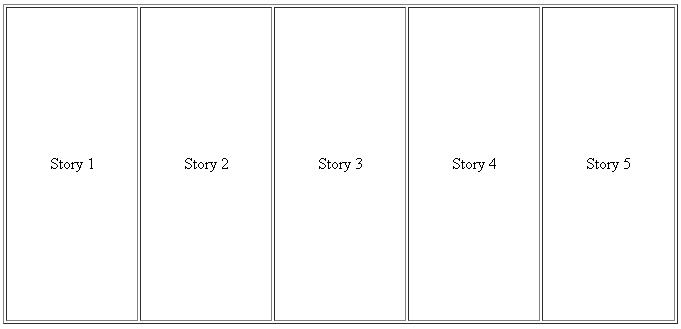
echo str\_replace("world", "Dolly", "Hello world!"); // outputs Hello Dolly!

?>

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

**VI. LABORATORY ACTIVITY**

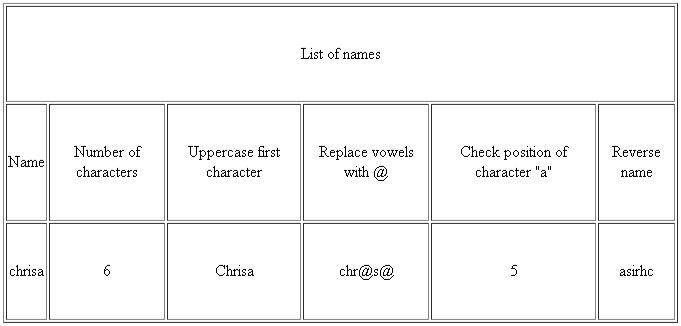
* 1. Create 5 different short story webpages and Convert your stories into a web based form, use include() and require() functions to connect all pages integrate with HTML and CSS (Put images on the stories). See the format below:



**Applications Development and Emerging Technologies** **Page 9 of 11**

1. Creation of String Function in PHP, create an array with 20 different names and get the following:
   * Number of characters of each name include spaces
   * Change the first character of each name into upper case
   * Replace all vowels with @ symbol
   * Check the position of letter a
   * Change the name into reverse formate

Please see the sample below



***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.***

**VII. QUESTION AND ANSWER**

1. **What is Constant in PHP**

A Constant is an identifier for a simple value which means that the value cannot be changed or altered during the script. In PHP, we use the define() function to create a constant.

1. **What is the difference between include and require? Explain**

The difference between include and require is that, include still runs the program even if the file you are accessing is not found. Meanwhile, the require will stop the program and display the error message and won’t continue to output the rest of the codes.

1. **What are the different string functions? Explain each**

First, is strlen(), which returns the length of the string. Next is str\_word\_count() which returns the number of words in a sentence or phrase. strrev() reverses the string, strpos() searches for a specific text inside the string. Lastly, str\_replace() replaces some characters with some other characters in a string depending on the parameters stated.

**VIII. REFERENCES**

1. <https://www.w3schools.com/css/>
2. <https://www.w3schools.com/html/>
3. <https://www.w3schools.com/php/php_variables.asp>
4. <https://www.w3schools.com/php/php_constants.asp>
5. <https://www.w3schools.com/php/php_includes.asp>
6. <https://www.w3schools.com/php/php_string.asp>



**Applications Development and Emerging Technologies** **Page 10 of 11**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 11 of 11**