ACTIVITY ANSWER SHEET

Name	Denzel T. Lanzaderas
Section:	BSIT 3R1

Instructions:

1. Push your output on your GITHUB repository.
2. Use the answer sheet provided save it as PDF file then push it to your GitHub.
3. Answer the ff. problems write it on the answer sheet.
4. Late submissions will no longer be accepted.
5. Caught copying outputs of others will be given sanctions.
6. Failure to follow these instructions will be given sanctions.

Activity 1: Control Structures

1. Write down the syntax in PHP for the ff.

1. if	if (*condition*) {
	    code to be executed if condition is true;
	}

2. if…else	if (*condition*) {
	    code to be executed if condition is true;
	} else {
	    code to be executed if condition is false;
	}

3. if…else if…else	if (*condition*) {
	    code to be executed if this condition is true;
	} elseif (*condition*) {
	    code to be executed if this condition is true;
	} else {
	    code to be executed if all conditions are false;
	}

4. switch…case	switch (*n*) { case *label1:* *code to be executed if n=label1;* break; case *label2:* *code to be executed if n=label2;* break; case *label3:* *code to be executed if n=label3;* break; ... default: *code to be executed if n is different from all labels;* }

5. for loop	for (*init counter; test counter; increment counter*)
	{
	    code to be executed;
	}

6. do while loop	do {
	    code to be executed; }
	while (*condition is true*);

7. while loop	while (*condition is true*) {
	code to be executed;
	}

8. foreach loop	foreach ($*array* as $*value*) {
	code to be executed;
	}

9. break statement	if (condition)
	{
	Break// the break statement terminate the execution of the code_
	}

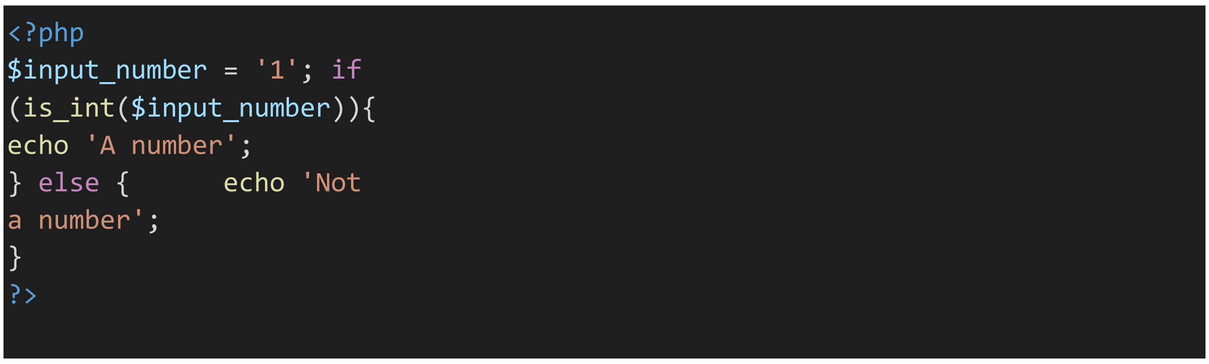
10. continue statement	if (condition)
	{
	continue;// *continue* accepts an optional numeric argument which tells it how many levels of enclosing loops it should skip to the end of
	}

11. try…catch	try
	{
	// run your code here
	}
	catch (exception $e) {
	//code to handle the exception
	}

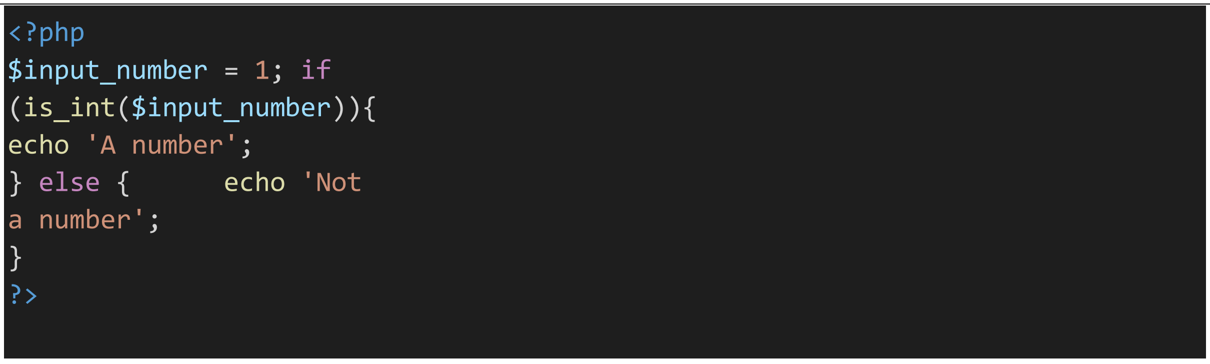
2. Solve the ff. problem using PHP.

a. Write a program that checks if value is a number (integer).

Sample input: ‘1’	Sample input: 1
Expected output: Not a number	Expected output: A number



```
<?php
$input_number = '1'; if
(is_int($input_number)){
echo 'A number';
} else {       echo 'Not
a number';
}
?>
```



```
<?php
$input_number = 1; if
(is_int($input_number)){
echo 'A number';
} else {       echo 'Not
a number';
}
?>
```

b. Write a program that checks if a value is positive or negative and odd or even.

Sample input: 0
Expected output: Positive & Even

Sample input: -1
Expected output: Negative and Odd

```
<?php
$input_number = 0; //$input_number = -1; function
oddandeven(){     global $input_number;     if
($input_number % 2 == 0 || $input_number==0){
echo "Even";
    }
else{
        echo "Odd";
    } } function
posandneg(){     global
$input_number;     if
($input_number >= 0){
echo 'Positive';
    }     else {
echo "Negative";
    } }
posandneg();
echo ' and ';
oddandeven();
?>
```

c. Write a program that checks if a value is palindrome.

Sample input: Anna	Sample input: Bogart
Expected output: Palindrome	Expected output: Not a Palindrome

```
<?php
$input_name = strtolower('Anna');
//$input_name = ('Bogart');
 if ($input_name ==
strrev($input_name)){     echo
"Palindrome";
} else {     echo "Not a
Palindrome";
}

?>
```

d. Write a program to calculate and print the factorial of a number using a for loop.

Sample input: 4

Expected output: 24

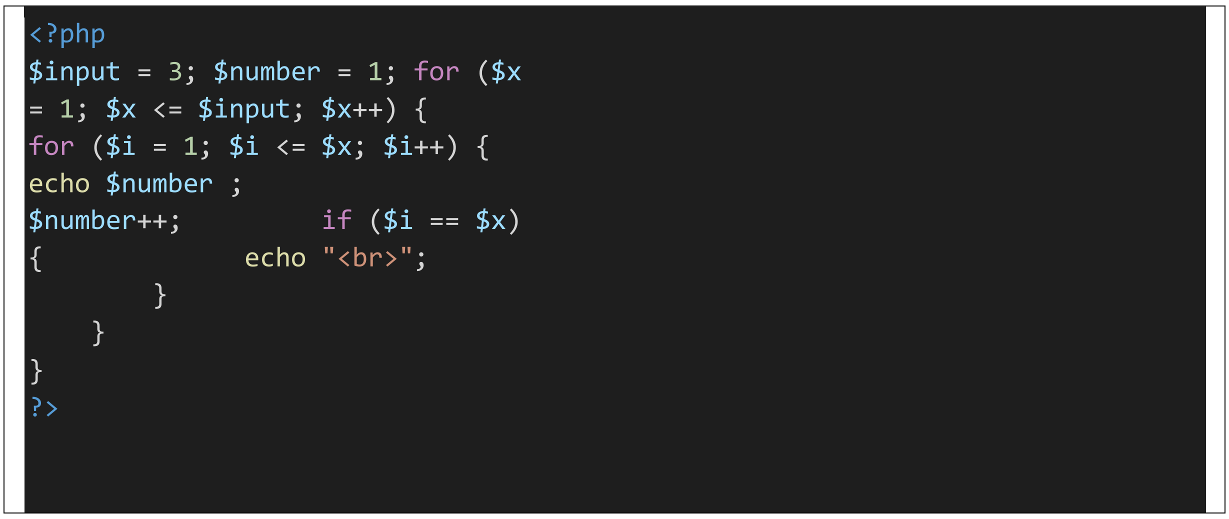
```
<?php
$number = 4;
$limit = $number - 1; for ($x =
1; $x <= $limit; $x++) {
    $number = $number * $x;
} echo
$number; ?>
```

e. Write a PHP program to generate and display the first n lines of a Floyd triangle.

Sample input: 3

Sample output:

```
1
2 3
4 5 6
```



```
<?php
$input = 3; $number = 1; for ($x
= 1; $x <= $input; $x++) {
for ($i = 1; $i <= $x; $i++) {
echo $number ;
$number++;         if ($i == $x)
{             echo "<br>";
        }
    }
}
?>
```

Activity 2: PHP Built-in Functions

Write down the functionalities of the ff. built-in functions in PHP.

Array	array(); Creates an array
Calendar	cal_days_in_month() Returns the number of days in a month for a specified year and calendar
Date	The date/time functions allow you to get the date and time from the server where your PHP script runs. You can then use the date/time functions to format the date and time in several ways.

Directory	The directory functions allow you to retrieve information about directories and their contents.
Error	The error functions are used to deal with error handling and logging. The error functions allow us to define own error handling rules, and modify the way the errors can be logged. The logging functions allow us to send messages directly to other machines, emails, or system logs. The error reporting functions allow us to customize what level and kind of error feedback is given.
File System	The filesystem functions allow you to access and manipulate the filesystem.
Filter	This PHP filters is used to validate and filter data coming from insecure sources, like user input.
FTP	The FTP functions give client access to file servers through the File Transfer Protocol (FTP). The FTP functions are used to open, login and close connections, as well as upload, download, rename, delete, and get information on files from file servers. Not all of the FTP functions will work with every server or return the same results. The FTP functions became available with PHP 3. If you only wish to read from or write to a file on an FTP server, consider using the ftp:// wrapper with the Filesystem functions which provide a simpler and more intuitive interface.
Libxml	The libxml functions and constants are used together with SimpleXML, XSLT and DOM functions.
Mail	The mail() function allows you to send emails directly from a script.

Math	The math functions can handle values within the range of integer and float types.

Misc	The misc. functions were only placed here because none of the other categories seemed to fit. The misc. functions were only placed here because none of the other categories seemed to fit.
MySQLi	The MySQLi functions allows you to access MySQL database servers.
Network	The Network functions contains various network function and let you manipulate information sent to the browser by the Web server, before any other output has been sent.
SimpleXML	SimpleXML is an extension that allows us to easily manipulate and get XML data. SimpleXML provides an easy way of getting an element's name, attributes and textual content if you know the XML document's structure or layout. SimpleXML turns an XML document into a data structure you can iterate through like a collection of arrays and objects.
Stream	The Stream functions .... Streams are the way of generalizing file, network, data compression, and other operations which share a common set of functions and uses. In its simplest definition, a stream is a resource object which exhibits streamable behavior. That is, it can be read from or written to in a linear fashion, and may be able to fseek() to an arbitrary location within the stream. A wrapper is additional code which tells the stream how to handle specific protocols/encodings.
String	The PHP string functions are part of the PHP core.

XML Parser	translates XML an XML document into a DOM tree-structure like document. CDATA is used to ignore special characters when parsing XML documents. PHP DOMDocument class to create XML files

Zip	The Zip files functions allows you to read ZIP files.
Timezones	used to set the default timezone used by all date/time functions in a script. This function returns False if the timezone is not valid, or True otherwise.

Activity 3: Regular Expression

1. Define Regular Expression (RegEx) and provide example programming scenario where you can use (RegEx). Provide example syntax in PHP.

Regular expressions are nothing more than a sequence or pattern of characters itself. They provide the foundation for pattern-matching functionality.

Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.

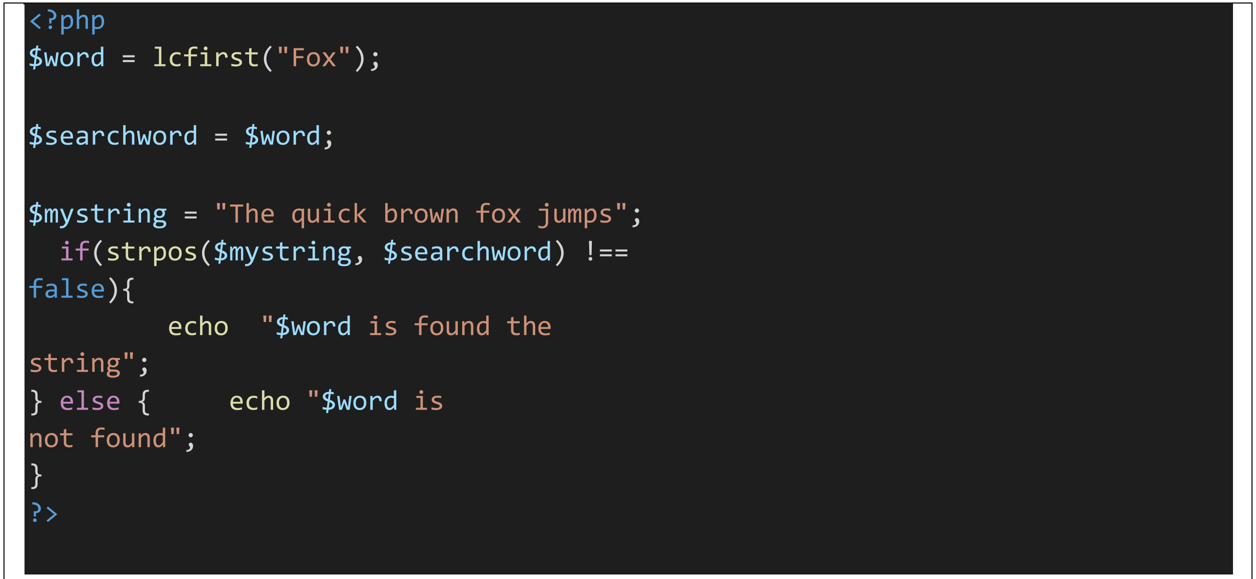
PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.

```
<?php
$string = 'The quick brown [fox].'; preg_match('#\[(.*?)\]#', $string, $same);
print $same[1]."\n";
?>
```

2. Solve the ff. problem using Regular Expressions.

a. Write a PHP script that checks if a string contains another string

Sample String: 'The quick brown fox'
Test input: ‘Fox’
Expected output: Fox is found the string



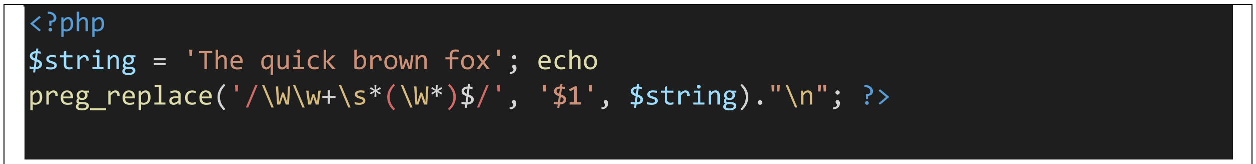
```
<?php
$word = lcfirst("Fox");

$searchword = $word;

$mystring = "The quick brown fox jumps";
  if(strpos($mystring, $searchword) !==
false){
         echo  "$word is found the
string";
} else {     echo "$word is
not found";
}
?>
```

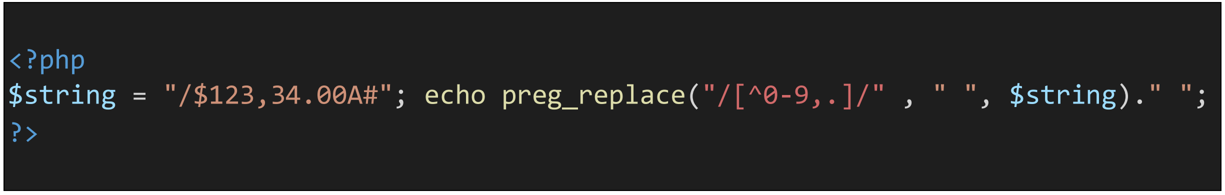
b. Write a PHP script that removes the last word from a string.

Sample String: 'The quick brown fox'
Expected output: ‘The quick brown'



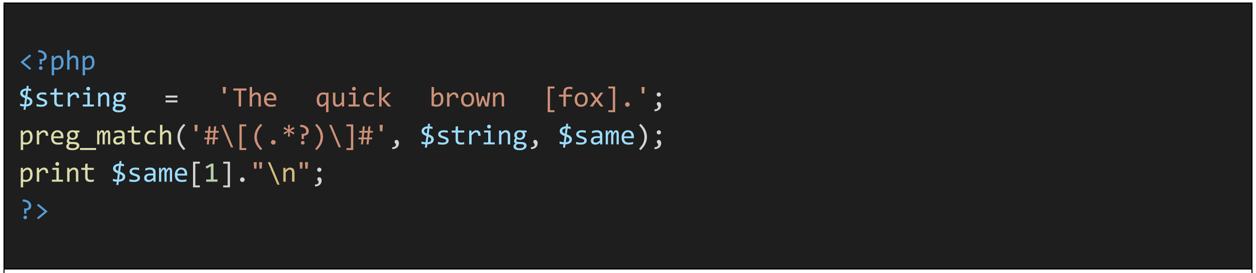
```php
<?php
$string = 'The quick brown fox'; echo
preg_replace('/\W\w+\s*(\W*)$/', '$1', $string)."\n"; ?>
```

c. Write a PHP script to remove nonnumeric characters except comma and dot.
Sample String: '/$123,34.00A#'
Expected output: 123,34.00



```php
<?php
$string = "/$123,34.00A#"; echo preg_replace("/[^0-9,.]/" , " ", $string)." ";
?>
```

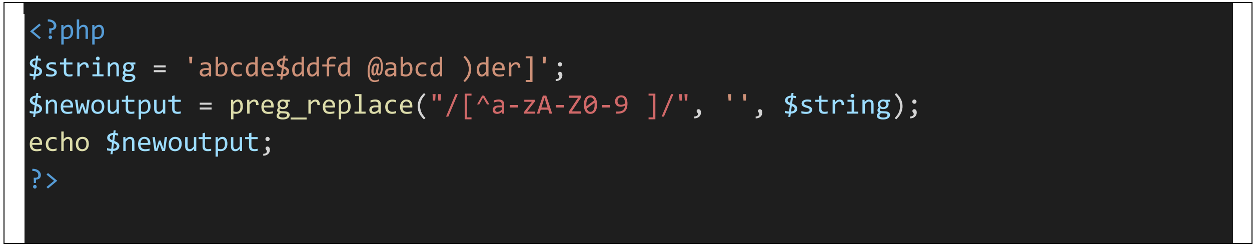
d. Write a PHP script to extract text (within parenthesis) from a string. Sample String: 'The quick brown [fox].'
Expected output: Fox



```php
<?php
$string  =  'The  quick  brown  [fox].';
preg_match('#\[(.*?)\]#', $string, $same);
print $same[1]."\n";
?>
```

e. Write a PHP script to remove all characters from a string except a-z A-Z 0-9 or " ".

Sample String: 'abcde$ddfd @abcd)der]'
Expected output: abcdeddfd abcd der



```php
<?php
$string = 'abcde$ddfd @abcd )der]';
$newoutput = preg_replace("/[^a-zA-Z0-9 ]/", '', $string);
echo $newoutput;
?>
```

Activity 4: Error Handling

1. List down the different PHP errors. Provide example code on how to handle these errors.

```
<?php     if($denzel
== 'sad')      echo
$output;
```

Parse error: syntax error, unexpected '}', expecting end of file in **C:\xampp\htdocs\index.php** on line **4**

Add { to the left parenthesis

```
<?php      if($denzel == 'sad'){     echo $output;
}
```

2.

```
<?php      if($denzel
== 'sad'){     echo
$output;
}
```

Notice: Undefined variable: denzel in **C:\xampp\htdocs\index.php** on line **3**

declare a value on variable $denzel $ output.

```
<?php
$denzel="sad"; $output='Lol' if($denzel == 'sad'){     echo $output;
}
```

Parse error: syntax error, unexpected 'if' (T_IF) in C:\xampp\htdocs\index.php on line 4

Add ?> at the end as container of the codes.

3.

```
<?php
$denzel="sad"; $output='Lol'; if($denzel == 'sad'){     echo $output;
}
?>
```