ACTIVITY ANSWER SHEET

Name	Christian P. Daohog
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) {`
	`  do something if condition is true;`
	`}`

2. if…else	`if (condition) {`
	`  do something if condition is true;`
	`} else {`
	`  do something if condition is false;`
	`}`

3. if…else if…else	`if (condition) {`
	`  do something if this condition is true;`
	`} elseif (condition) {`
	`  do something if first condition is false and this condition is true;`
	`} else {`
	`  do something 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 for each iteration;`
	`}`

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	`break;`
10. continue statement	`continue;`

11. try…catch	`// code before the try-catch block`
	
	`try {`
	`// code`
	
	`// if something is not as expected`
	`// throw exception using the "throw" keyword`

```
  // code, it won't be executed if the above exception is thrown
} catch (Exception $e) {
  // exception is raised and it'll be handled here
  // $e->getMessage() contains the error message
}

// code after the try-catch block, will always be executed
}
```

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
<?php
    $num=1;
    $letter="1";
    if(is_int($letter)) {
        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 Sample input: -1
Expected output: Positive & Even Expected output: Negative and Odd

```php
<?php
    function integer_check ($num) {
        if ($num>=0) {
            odd_even($num, "Positive");
        }
        else {
            odd_even($num, "Negative");
        }
    }

    function odd_even($num, $sign){
        if ($num%2==0){
            echo $sign." & Even <br>";
        }
        else{
            echo $sign." & Odd <br>";
        }
    }

integer_check(0);
integer_check(-1);
?>
```

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

Sample input: Anna
Expected output: Palindrome

Sample input: Bogart
Expected output: Not a Palindrome

```php
<?php
    function palindrome ($word){
        if (strrev($word) == $word) {
            echo "Palindrome <br>";
        }
        else {
            echo "Not a Palindrome <br>";
        }
    }

palindrome(strtolower("Anna"));
palindrome(strtolower("bogart"));
?>
```

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

Sample input: 4
Expected output: 24

```php
<form action="index.php" method="post">
Input:  <input type="text" name="name"><br>
<input type="submit" style="position: absolute; left: 165px; margin-top:5px">
</form>

<?php
    $num = $_POST["name"];
    $facto=1;
    for ($initial_val = 1; $initial_val<=$num; $initial_val++){
        $facto *= $initial_val;
    }
    echo $num.'! = '.$facto;
?>
```

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
<form action="index.php" method="post">
Input:  <input type="text" name="name"><br>
<input type="submit" style="position: absolute; left: 165px; margin-top:5px">
</form>

<?php
    $num = $_POST["name"];
    $incre=1;
    for ($initial=1; $initial<=$num; $initial++){
        for($second_initial=1; $second_initial<=$initial; $second_initial++){
            echo $incre.' ';
            $incre++;
        }
        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
	array_change_key_case()	- Changes all keys in an array to lowercase or uppercase
	array_chunk()	- Splits an array into chunks of arrays
	array_column()	- Returns the values from a single column in the input array
	array_combine()	- Creates an array by using the elements from one "keys" array and one "values" array
Calendar	**cal_days_in_month()**	- Returns the number of days in a month for a specified year and calendar
	cal_from_jd()	- Converts a Julian Day Count into a date of a specified calendar
	cal_info()	- Returns information about a specified calendar
	cal_to_jd()	- Converts a date in a specified calendar to Julian Day Count
	easter_date()	- Returns the Unix timestamp for midnight on Easter of a specified year
Date	**checkdate()**	- Validates a Gregorian date
	date_add()	- Adds days, months, years, hours, minutes, and seconds to a date

	date_create_from_format()	- Returns a new DateTime object formatted according to a specified format
	date_create()	- Returns a new DateTime object
	date_date_set()	- Sets a new date
Directory		
	checkdate()	- Validates a Gregorian date
	date_add()	- Adds days, months, years, hours, minutes, and seconds to a date
	date_create_from_format()	- Returns a new DateTime object formatted according to a specified format
	date_create()	- Returns a new DateTime object
	date_date_set()	- Sets a new date
Error	**debug_backtrace()**	- Generates a backtrace
	debug_print_backtrace()	- Prints a backtrace
	error_clear_last()	- Clears the last error
	error_get_last()	- Returns the last error that occurred
	error_log()	-Sends an error message to a log, to a file, or to a mail account
File System	**basename()**	- Returns the filename component of a path
	chgrp()	- Changes the file group
	chmod()	- Changes the file mode
	chown()	- Changes the file owner
	clearstatcache()	-Clears the file status cache

Filter	**filter_has_var()**	- Checks whether a variable of a specified input type exist
	filter_id()	- Returns the filter ID of a specified filter name
	filter_input()	- Gets an external variable (e.g. from form input) and optionally filters it
	filter_input_array()	- Gets external variables (e.g. from form input) and optionally filters them
	filter_list()	- Returns a list of all supported filter names
FTP	**ftp_alloc()**	- Allocates space for a file to be uploaded to the FTP server
	ftp_cdup()	- Changes to the parent directory on the FTP server
	ftp_chdir()	- Changes the current directory on the FTP server
	ftp_chmod()	- Sets permissions on a file via FTP
	ftp_close()	- Closes an FTP connection
Libxml	**libxml_clear_errors()**	- Clears the libxml error buffer
	libxml_disable_entity_loader()	- Enables the ability to load external entities
	libxml_get_errors()	- Gets the errors from the the libxml error buffer
	libxml_get_last_error()	- Gets the last error from the the libxml error buffer
	libxml_set_external_entity_loader()	- Changes the default external entity loader

Mail	**ezmlm_hash()**	- Calculates the hash value needed by EZMLM
	mail()	- Allows you to send emails directly from a script
Math	**abs()**	- Returns the absolute (positive) value of a number
	acos()	- Returns the arc cosine of a number
	acosh()	- Returns the inverse hyperbolic cosine of a number
	asin()	- Returns the arc sine of a number
	asinh()	- Returns the inverse hyperbolic sine of a number
Misc	**connection_aborted()**	- Checks whether the client has disconnected
	connection_status()	- Returns the current connection status
	constant()	- Returns the value of a constant
	define()	- Defines a constant
	defined()	- Checks whether a constant exists
MySQLi	**affected_rows()**	- Returns the number of affected rows in the previous MySQL operation
	autocommit()	- Turns on or off auto-committing database modifications
	begin_transaction()	- Starts a transaction
	change_user()	- Changes the user of the specified database connection
	character_set_name()	- Returns the default character set for the database connection
Network	**checkdnsrr()**	- Checks DNS records for type corresponding to host

	closelog()	- Closes the connection of system logger
	dns_check_record()	- Alias of **checkdnsrr()**
	dns_get_mx()	- Alias of **getmxrr()**
	dns_get_record()	- Gets the DNS resource records associated with the specified hostname
SimpleXML	**__construct()**	- Creates a new SimpleXMLElement object
	__toString()	- Returns the string content of an element
	addAttribute()	- Appends an attribute to the SimpleXML element
	addChild()	- Appends a child element the SimpleXML element
	asXML()	- Returns a well-formed XML string (XML version 1.0) from a SimpleXML object
Stream	**stream_bucket_prepend()**	
	stream_context_create()	
	stream_context_get_default()	
	stream_context_get_options()	
	stream_context_get_params()	
String	**addcslashes()**	- Returns a string with backslashes in front of the specified characters
	addslashes()	- Returns a string with backslashes in front of predefined characters

	bin2hex()	- Converts a string of ASCII characters to hexadecimal values
	chop()	- Removes whitespace or other characters from the right end of a string
	chr()	- Returns a character from a specified ASCII value
XML Parser	**utf8_decode()**	- Decodes an UTF-8 string to ISO-8859-1
	utf8_encode()	- Encodes an ISO-8859-1 string to UTF-8
	xml_error_string()	- Returns an error string from the XML parser
	xml_get_current_byte_index()	- Returns the current byte index from the XML parser
	xml_get_current_column_number()	- Returns the current column number from the XML parser
Zip	**zip_close()**	- Closes a ZIP file archive
	zip_entry_close()	- Closes a ZIP directory entry
	zip_entry_compressedsize()	- Returns the compressed file size of a ZIP directory entry
	zip_entry_compressionmethod()	- Returns the compression method of a ZIP directory entry
	zip_entry_filesize()	- Returns the actual file size of a ZIP directory entry
Timezones	**Africa** **America** **Antarctica** **Arctic** **Asia**	

Activity 3: Regular Expression

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

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

```
<form action="index.php" method="post">
Input:  <input type="text" name="name"><br>
<input type="submit" style="position: relative; left: 156px; margin-top:5px">
</form>

<?php
$pattern = $_POST["name"];
$text = "The quick brown fox";
if(preg_match("/$pattern/", $text)){
    echo "$pattern is found the string";
} else{
    echo "$pattern is not found in the string";
}
?>
```

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
$my_text="The quick brown fox";
$text = preg_replace('/\W\w+\s*(\W*)$/', '$1', $my_text);
Echo $text;
?>
```

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

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

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

```php
<?php
$my_text = 'The quick brown [fox].';
$text = preg_match('#\[(.*?)\]#', $my_text, $match);
echo $match[1];
?>
```

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
$my_text = 'abcde$ddfd @abcd )der]';
$text = preg_replace("/[^A-Za-z0-9 ]/", '', $my_text);
echo $text;
?>
```

Activity 4: Error Handling

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

A. If the file does not exist you might get an error like this:

```
Warning: fopen(mytestfile.txt) [function.fopen]: failed to

open stream:

No such file or directory in C:\webfolder\test.php on line 2
```

Handling errors:

```php
<?php
if(file_exists("mytestfile.txt")) {
  $file = fopen("mytestfile.txt", "r");
} else {
  die("Error: The file does not exist.");
}
?>
```

B. trigger_error() Function

```
Notice: Value must be 1 or below

in C:\webfolder\test.php on line 6
```

```php
<?php

$test=2;

if ($test>=1) {

  trigger_error("Value must be 1 or below");

}

?>
```