ACTIVITY ANSWER SHEET

Name	Patlunag, Rey L.
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;
	} else {
	    code to be executed if condition is false;
	}

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

3. if…else if…else	if (condition) {
	    code to be executed if this condition is true;
	} elseif (condition) {
	    code to be executed if first condition is false and 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 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	function checkNum($number) {
	  if($number>1) {
	    throw new Exception("Value must be 1 or below");
	  }
	  return true;
	}
	
	//trigger exception in a "try" block
	try {
	  checkNum(2);
	  //If the exception is thrown, this text will not be shown
	  echo 'If you see this, the number is 1 or below';
	}
	
	//catch exception
	catch(Exception $e) {
	  echo 'Message: ' .$e->getMessage();
	}

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
$mynumber = 1;
$mystring = '1';
if (is_int($mynumber)) {
   echo "$mynumber is integer\n";
} else {
   echo "$mynumber is not an integer";
}
echo "<br>";

if (is_int($mystring)) {
   echo "$mystring is integer\n";
} else {
   echo "$mystring is not an integer <br> ";
}

?>
```

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
$mynumber = 2;
if ($mynumber >= 0){
        if ($mynumber % 2 == 0){
        echo "$mynumber is Positive and Even";
    echo "<br>";
  }
  else{
        echo "$mynumber is Positive and Odd";
    echo "<br>";
  }
}
else{
        if ($mynumber % 2 == 0){
        echo "$mynumber is Negative and Even";
    echo "<br>";
  }
  else{
        echo "$mynumber is Negative and Odd";
    echo "<br>";
  }
}
?>
```

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
function test_palindrome($mystring)
{
 if ($mystring == strrev($mystring))
   return 1;
 else
        return 0;
}
$test1 = 'anna';
if (test_palindrome($test1)){
  echo "$test1 is a Palindrome";
}
else {
        echo "$test1 is not a Palindrome";
}
echo "<br>";
$test2 = 'bogart';
if (test_palindrome($test2)){
  echo "$test2 is a Palindrome";
}
else {
        echo "$test2 is 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
function calc_factorial($mynumber){
   $factorial = 1;
   for ($x = 1; $x <= $mynumber; $x++){
     $factorial = $factorial * $x;
   }
   return $factorial;
}
$test = 4;
$factor = calc_factorial($test);
echo "Output = $factor";
?>
```

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
$mynumber = 3;
$counter = 1;
for ($x = $mynumber; $x > 0; $x--)
{
  for ($y = $x; $y < $mynumber + 1; $y++)
   {
     printf("%4s", $counter);
     $counter++;
   }
             echo "<br>";
   }
?>
```

Activity 2: PHP Built-in Functions

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

Array	array() is an inbuilt **function** in **PHP** which is used to create an **array**.
	
	array_column()
	array_combine()
	array_count_values ()
	array_diff0()
	array_fill()

Calendar	functions that simplifies converting between different calendar formats
	
	cal_days_in_month()
	cal_from_jd()
	cal_info()
	cal_to_jd()
	easter_date()

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.
	
	checkdate()
	date_add()
	date_create_from_format()
	date_create()
	date_date_set()

Directory	The directory functions allow you to retrieve information about directories and their contents.
	
	chdir()
	chroot()
	closedir()
	dir()
	getcwd()

Error	The error functions are used to deal with error handling and logging.
	
	debug_backtrace()
	debug_print_backtrace()
	error_clear_last()
	error_get_last()
	error_log()

File System	The filesystem functions allow you to access and manipulate the filesystem.
	
	basename()
	chgrp()
	chmod()
	chown()
	clearstatcache()

Filter	This PHP filters is used to validate and filter data coming from insecure sources, like user input.
	
	filter_has_var()
	filter_id()
	filter_input()
	filter_input_array()
	filter_list()

FTP	The FTP functions give client access to file servers through the File Transfer Protocol (FTP).
	
	ftp_alloc()
	ftp_cdup()
	ftp_chdir()
	ftp_chmod()
	ftp_close()

Libxml	The libxml functions and constants are used together with SimpleXML, XSLT and DOM functions.
	
	libxml_clear_errors()
	libxml_disable_entity_loader()
	libxml_get_errors()
	libxml_get_last_error()
	libxml_set_external_entity_loader()

Mail	The mail() function allows you to send emails directly from a script.
	
	ezmlm_hash()
	mail()

Math	The math functions can handle values within the range of integer and float types.
	
	abs()
	acos()
	acosh()
	asin()
	asinh()

Misc	The misc. functions were only placed here because none of the other categories seemed to fit.
	
	connection_aborted()
	connection_status()
	connection_timeout()
	constant()
	define()

MySQLi	The MySQLi functions allows you to access MySQL database servers.
	
	affected_rows()
	autocommit()
	begin_transaction()
	change_user()
	character_set_name()

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.
	
	checkdnsrr()
	closelog()
	define_syslog_variables()
	dns_check_record()
	Alias of checkdnsrr()

SimpleXML	SimpleXML is an extension that allows us to easily manipulate and get XML data.
	
	__construct()
	__toString()
	addAttribute()
	addChild()
	asXML()

Stream	Streams are the way of generalizing file, network, data compression, and other operations which share a common set of functions and uses.
	
	stream_bucket_prepend()
	stream_context_create()
	stream_context_get_default()
	stream_context_get_options()
	stream_context_get_params()

String	The PHP string functions are part of the PHP core. No installation is required to use these functions.
	
	addcslashes()
	addslashes()
	bin2hex()
	chop()
	chr()

XML Parser	The XML parser functions lets you create XML parsers and define handlers for XML events.
	
	utf8_decode()
	utf8_encode()
	xml_error_string()
	xml_get_current_byte_index()
	xml_get_current_column_number()

Zip	The Zip files functions allows you to read ZIP files.
	
	zip_close()
	zip_entry_close()
	zip_entry_compressedsize()
	zip_entry_compressionmethod()
	zip_entry_filesize()

Timezones	date_default_timezone_get()
	date_default_timezone_set(timezone)
	date_timezone_get(object)
	date_timezone_set(object, timezone)
	timezone_version_get()

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 powerful pattern matching algorithm that can be performed in a single expression.

When creating a custom HTML template. Regular expressions can be used to identify the template tags and replace them with actual data.

Example syntax in PHP:

```
<?php
function_name('/pattern/',subject);
?>
```

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
$mystring = "The quick brown fox";
$test = "/Fox/i";
if (preg_match($test, $mystring))
{
	echo "Fox is found in the string";
}
else
{
	echo "Fox 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
$mystring = "The quick brown fox";
echo preg_replace('/\W\w+\s*(\W*)$/', '$1', $mystring)."\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
$mystring = "/$123,34.00A#";
echo preg_replace("/[^0-9,.]/", "", $mystring)."\n";
?>
```

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

```
<?php
$mystring = 'The quick brown [fox].';
preg_match('#\[(.*?)\]#', $mystring, $matched);
print $matched[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
$mystring = 'abcde$ddfd @abcd  )der]';
$test = preg_replace("/[^A-Za-z0-9 ]/", '', $mystring);
echo 'Output : '.$test."\n";
?>
```

Activity 4: Error Handling

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

Parse error or Syntax Error: It is the type of error done by the programmer in the source code of the program.

```
<?php include("csharpcorner.php"); ?>
```

Fatal Error: It is the type of error where PHP compiler understand the PHP code but it recognizes an undeclared function.

```
function shutDownFunction() {
    $error = error_get_last();
    // fatal error, E_ERROR === 1
    if ($error['type'] === E_ERROR) {
        //do your stuff
    }
}
register_shutdown_function('shutDownFunction');
```

Warning Errors : The main reason of warning errors are including a missing file. This means that the PHP function call the missing file.

Notice Error: It is similar to warning error. It means that the program contains something wrong but it allows the execution of script.

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