ACTIVITY ANSWER SHEET

Name	Richie Marie S. Gatela
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 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 (condiion 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	try {
	
	}
	catch (exception $e) {
	
	}

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

function Integer($s)

{

    for ($i = 0; $i < strlen($s); $i++)

        if (is_numeric($s[$i]) == false)

            return false;

    return true;

}

$Input = 1;

if (Integer($Input))

    echo "Not a number";

else

    echo "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 check($value){

    if($value % 2 == 0){

        echo "Positive & Even";

    }

    else{

        echo "Negative & Odd";

    }
```

```
}

$Input = 0;

check($Input)

?>
```

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

function checkPal( $str1)

{

    $str1 = str_replace( ' ', '', $str1 );

    return $str1 == strrev( $str1 );

}

$word = 'anna';

if( checkPal( $word ) == true )

{

    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

$input = 4;

$loop = 1;

for($r=1;$r<=$input-1;$r++)

{

 $loop*=($r+1);

}

echo "$loop"."\n";

?>
```

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
<?php

$input = 3;

echo "\n";

$value = 1;

for ($i = $input; $i > 0; $i--)

{

  for ($j = $i; $j < $input + 1; $j++)

   {

     printf("%4s", $value);

     $value++;

   }

  echo "\n";

   }

?>
```

Activity 2: PHP Built-in Functions

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

Array	Allow you to access and manipulate arrays.
Calendar	Simplifies converting between different calendar formats.
Date	Simplify working with date data types. Used to format a date or time into a human readable format.
Directory	Allow you to retrieve information about directories and their contents.
Error	Used to deal with error handling and logging.
File System	Allow you to access and manipulate the filesystem.
Filter	Used to validate and filter data coming from insecure sources, like user input.
FTP	Give client access to file servers.
Libxml	Used together with SimpleXM:, XSLT and DOM functions.
Mail	Allow you to send emails directly from a script.

Math	Can handle values within the range of integer and float types.
Misc	Were only placed here because none of the other categories seemed to fit.
MySQLi	Allows you to access MYSQL database servers.
Network	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	An extension that allows us to easily manipulate and get XML data.
Stream	The way of generalizing file, network, data compression, and other operations which share a common set of functions and uses.
String	Used in computer programming languages to manipulate a string or query information about a string (some do both).
XML Parser	Let you create XML parsers and define handlers for XML events.
Zip	Allows you to read ZIP files.
Timezones	Used to set the default timezone used by all date/time functions in a script.

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 Expression (RegEx) are a specially formatted text strings used to find patterns in text. An effective and efficient text processing and manipulations.

```php
<?php

$pattern = "/c[ao]pe/";

$text = "Two words cape and cope";

$matches = preg_match_all($pattern, $text, $array);

echo $matches . " matches were found."

?>
```

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
<?php

$string = "The quick brown fox";

if (strpos($string, 'fox') !== false)

    {

        echo 'Fox is found in the string.';

    }

else

    {

        echo 'Fox in 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
<?php

$my_string = 'The quick brown fox';

echo implode(' ', array_slice(explode(' ',$my_string), 0, 3)). "\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

$nonnumeric = "'/$123,34.00A#'";

echo preg_replace("/[^0-9,.]/", "", $nonnumeric)."\n";

?>
```

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

```php
<?php

$str1 = 'The quick brown [Fox].';

preg_match('#\[(.*?)\]#', $str1, $string);

print $string[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

$script = 'abcde$ddfd @abcd )der]';

$newstring = preg_replace("/[^a-zA-Z0-9 ]/", '', $script);

echo $newstring."\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**

```php
<?php

$input = 3;

echo "\n";

$value = 1;

for ($i = $input; $i > 0; $i--)

{

  for ($j = $i; $j < $input + 1; $j++)

   {

     printf("%4s", $value)
```

```
    $value++;
   }
  echo "\n"
   }
?>
```

*** Fatal Error:**

```
<?php
function sub()
{
$sub=6-1;
echo "The sub= ".$sub;
}
div();
?>
```

*** Warning Errors:**

```
<?php

$x = "GeeksforGeeks";

include ("gfg.php");

echo $x . "Computer science portal";
?>
```

*** Notice Error:**

```
<?php

$x = "GeeksforGeeks";

echo $x;

echo $geeks;
?>
```