

ACTIVITY ANSWER SHEET

Name	Nida D. Mandawe
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	<pre>if (condition) { code to be executed if condition is true; }</pre>
2. if...else	<pre>if (condition) { code to be executed if condition is true; } else { code to be executed if condition is false; }</pre>
3. if...else if...else	<pre>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; }</pre>
4. switch...case	<pre>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; }</pre>
5. for loop	<pre>for (init counter; test counter; increment counter) { code to be executed; }</pre>
6. do while loop	<pre>do { code to be executed; } while (condition is true);</pre>
7. while loop	<pre>while (condition is true) { code to be executed; }</pre>
8. foreach loop	<pre>foreach (\$array as \$value) { code to be executed; }</pre>

9. break statement	While (condition is true) { Code to be executed; } Break;
10. continue statement	For (condition is true){ Code to be executed; Continue; }
11. try...catch	<pre>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(); }</pre>

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

```
$num = 1;  
  
if (is_integer($num)){  
    echo $num, ' ', "This is a number";  
  
else {  
    echo $num, ' ', "This is a String 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  
$number = -3;  
  
if ($number>=0){  
    if ($number %2 == 0){  
        echo "positive and Even";  
    }  
    else  
        echo "positive and Odd";  
}  
else{  
    if ($number <= 0){  
        if ($number %2 == 0){  
            echo "Negative and even";  
        }  
        else {
```

```

        echo "Negative and odd";
    }
}
}

?>

```

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
    $word = "Prisy";
    if (strrev($word) == $word){
        echo $word, ' ', "is Palindrome";
    }
    else{
        echo $word, ' ', "is not 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 c_factor($mynum1){
        $factorial = 1;
        for ($num = 1; $num <= $mynum1; $num++){
            $factorial = $factorial * $num;
        }
        return $factorial;
    }

    $given = 5;
    $fact = c_factor($given);
    echo "Output = $fact";

?>

```

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;
$count = 1;
for ($next = $mynumber; $next > 0; $next--)
{
    for ($next1 = $next; $next1 < $mynumber + 1; $next1++)
    {
        printf("%4s", $count);
        $count++;
    }
    echo "<br>";
}

?>

```

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. array_fill_keys() Fills an array with values, specifying keys array_filter() Filters the values of an array using a callback function array_flip() Flips/Exchanges all keys with their associated values in an array array_intersect() Compare arrays, and returns the matches (compare values only) array_intersect_assoc() Compare arrays and returns the matches (compare keys and values)
Calendar	- contains functions that simplifies converting between different calendar formats. 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	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() 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	allow you to retrieve information about directories and their contents. chdir() Changes the current directory chroot() Changes the root directory closedir() Closes a directory handle dir() Returns an instance of the Directory class getcwd() Returns the current working directory
Error	used to deal with error handling and logging. allow us to define own error handling rules, and modify the way the errors can be logged. debug_backtrace() Generates a backtrace debug_print_backtrace() Prints a backtrace 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 error_reporting() Specifies which errors are reported
File System	allow you to access and manipulate the filesystem. 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	is used to validate and filter data coming from insecure sources, like user input. FILTER_VALIDATE_BOOLEAN 258 Validates a boolean FILTER_VALIDATE_EMAIL 274 Validates an e-mail address FILTER_VALIDATE_FLOAT 259 Validates a float

	FILTER_VALIDATE_INT 257 Validates an integer FILTER_VALIDATE_IP 275 Validates an IP address
FTP	give client access to file servers through the File Transfer Protocol (FTP). ftp_chmod() Sets permissions on a file via FTP ftp_close() Closes an FTP connection ftp_connect() Opens an FTP connection ftp_delete() Deletes a file on the FTP server ftp_exec() Executes a command on the FTP server
Libxml	functions and constants are used together with SimpleXML, XSLT and DOM functions. LIBXML_DTDATTR Set default DTD attributes LIBXML_DTDLOAD Load external subset LIBXML_DTDVALID Validate with the DTD LIBXML_NOBLANKS Remove blank nodes LIBXML_NOCDATA Set CDATA as text nodes
Mail	allows you to send emails directly from a script. SMTP "localhost" Windows only: The DNS name or IP address of the SMTP server smtp_port "25" Windows only: The SMTP port number. For PHP 4.3.0 and above sendmail_from NULL Windows only: Specifies the "from" address to be used when sending mail from mail() ezmlm_hash() Calculates the hash value needed by EZMLM mail() Allows you to send emails directly from a script
Math	can handle values within the range of integer and float types. 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	were only placed here because none of the other categories seemed to fit. constant() Returns the value of a constant define() Defines a constant defined() Checks whether a constant exists die() Prints a message and exits the current script eval() Evaluates a string as PHP code
MySQLi	allows you to access MySQL database servers. mysqli_field_seek() Sets the field cursor to the given field offset mysqli_field_tell() Returns the position of the field cursor mysqli_free_result() Frees the memory associated with a result mysqli_get_charset() Returns a character set object mysqli_get_client_info() Returns the MySQL client library version
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. getprotobyname() Returns the protocol number for a given protocol name getprotobynumber() Returns the protocol name for a given protocol number getservbyname() Returns the port number for a given Internet service and protocol getservbyport() Returns the Internet service for a given port and protocol header_register_callback() Calls a header function
SimpleXML	an extension that allows us to easily manipulate and get XML data. Provides an easy way of getting an element's name, attributes and textual content if you know the XML document's structure

	<p>or layout.</p> <p>attributes() Returns the attributes/values of an element</p> <p>children() Returns the children of a specified node</p> <p>count() Counts the children of a specified node</p> <p>getDocNamespaces() Returns the namespaces DECLARED in document</p> <p>getName()Returns the name of the XML tag referenced by the SimpleXML element</p>
Stream	<p>way of generalizing file, network, data compression, and other operations which share a common set of functions and uses.</p> <p>stream_context_get_options()</p> <p>stream_context_get_params()</p> <p>stream_context_set_default()</p> <p>stream_context_set_options()</p> <p>stream_context_set_params()</p>
String	<p>part of the PHP core. No installation is required to use these functions.</p> <p>count_chars() Returns information about characters used in a string</p> <p>crc32() Calculates a 32-bit CRC for a string</p> <p>crypt() One-way string hashing</p> <p>echo() Outputs one or more strings</p> <p>explode() Breaks a string into an array</p>
XML Parser	<p>lets you parse, but not validate, XML documents.</p> <p>xml_parser_create() Create an XML parser</p> <p>xml_parser_free() Free an XML parser</p> <p>xml_parser_get_option() Get options from an XML parser</p> <p>xml_parser_set_option() Set options in an XML parser</p> <p>xml_set_character_data_handler() Set handler function for character data</p>
Zip	<p>allows you to read ZIP files.</p> <p>zip_entry_open() Opens an entry in the ZIP file for reading</p> <p>zip_entry_read() Reads from an open entry in the ZIP file</p> <p>zip_open() Opens a ZIP file</p> <p>zip_read() Reads the next entry in a ZIP file</p> <p>zip_close() Closes a ZIP file</p>
Timezones	<p>which are useful with several PHP date functions.</p> <p>Africa</p> <p>America</p> <p>Antarctica</p> <p>Arctic</p> <p>Asia</p>

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.

A word boundary character (\b) helps you search for the words that begins and/or ends with a pattern.

```
<?php
$pattern = '\bcar\b*';
$replacement = '<b>$0</b>';
$text = 'Words beginning with car: cart, carrot, cartoon. Words ending with car: scar, oscar, supercar.';
echo preg_replace($pattern, $replacement, $text);
?>
```

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
$string = "The quick brown fox";
$test = "/Fox/i";
if (preg_match($test, $string))
{
    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
$string = "The quick brown fox";
echo preg_replace('/\W\w+\s*(\W*)$/', '$1', $string);
?>
```

c. Write a PHP script to remove nonnumeric characters except comma and dot.

Sample String: '\$123,34.00A#'

Expected output: 123,34.00

```
<?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
$string = 'The quick brown [fox].';
preg_match('#\[(.*?)\]#', $string, $like);
print $like[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: abcdeddfdf abcd der

```
<?php
$letters = 'abcde$ddfd @abcd )der]';
$go = preg_replace("/[^A-Za-z0-9 ]/", ' ', $letters);
echo 'Output :', ' '.$go;
?>
```

Activity 4: Error Handling

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

- File does not exist

Warning: fopen(welcome.txt) [function.fopen]: failed to open stream:
No such file or directory in C:\webfolder\test.php on line 2

Sample Code:

```
<?php
if(!file_exists("welcome.txt")) {
die("File not found");
} else {
$file=fopen("welcome.txt","r");
}
?>
```

- Trigger an Error

In a script where users can input data it is useful to trigger errors when an illegal input occurs. In PHP, this is done by the `trigger_error()` function.

Sample Code:

```
<?php
$test=2;
if ($test>=1) {
    trigger_error("Value must be 1 or below");
}
?>
```

- E_USER_ERROR - Fatal user-generated run-time error. Errors that can not be recovered from. Execution of the script is halted.

Sample Code:

```
<?php
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Ending Script";
    die();
}

//set error handler
set_error_handler("customError",E_USER_WARNING);

//trigger error
$test=2;
if ($test>=1) {
    trigger_error("Value must be 1 or below",E_USER_WARNING);
}
?>
```

- By default, PHP sends an error log to the server's logging system or a file, depending on how the `error_log` configuration is set in the `php.ini` file. By using the `error_log()` function you can send error logs to a specified file or a remote destination.

Sample Code:

```
<?php
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Webmaster has been notified";
    error_log("Error: [$errno] $errstr",1,
        "someone@example.com","From: webmaster@example.com");
}

//set error handler
set_error_handler("customError",E_USER_WARNING);

//trigger error
$test=2;
```

```
if ($test>=1) {  
    trigger_error("Value must be 1 or below",E_USER_WARNING);  
}  
?>
```