

ACTIVITY ANSWER SHEET

Name	Shaina Mae V. Montareal
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 first condition is false and 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 for each iteration; }</pre>

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

```
<!DOCTYPE HTML>
<html>
<body>
  <form action="sample.php" method="get">
    Input here: <input type="text" name="value">
    <input type="submit">
  </form>
</body>
</html>
<?php

$Value = ($_GET['value']);

if(is_numeric($Value))
{
    if($Value%2 && $Value<0){
        echo "Negative & Odd";
    }elseif($Value%2 && $Value>0){
        echo "Positive & Odd";
    }elseif($Value<0){
        echo "Negative & Even";
    }else{
        echo "Positive & Even";
    }
}
else
{
    echo "Not a number" ;
}

?>
```

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

```
<!DOCTYPE HTML>
<html>
<body>
  <form action="sample.php" method="get">
    Input here: <input type="text" name="value">
    <input type="submit">
  </form>
</body>
</html>
<?php

$Value = ($_GET['value']);

$val = strrev($Value);

if($Value==$val){
    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

```
<!DOCTYPE HTML>
<html>
<body>
  <form action="sample.php" method="get">
    Input here: <input type="text" name="value">
    <input type="submit">
  </form>
</body>
</html>
<?php

$Value = ($_GET['value']);
$Factorial = 1;

for($i=1;$i<=$Value-1;$i++){
  $Factorial*=(i+1);
}
echo "$Factorial";

?>
```

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

```
<!DOCTYPE HTML>
<html>
<body>
  <form action="sample.php" method="get">
    Input here: <input type="text" name="value">
    <input type="submit">
  </form>
</body>
</html>
<?php

$Value = ($_GET['value']);
$v = $Value;
$count = 1;

for($i=$v; $i>0; $i--){
  for($j=$i; $j<$v+1;$j++){
    printf("%4s",$count);
    $count++;
  }
  echo "<br>";
}

?>
```

Activity 2: PHP Built-in Functions

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

Array	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_column() - Returns the values from a single column in the input 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_days() - Returns the number of days after March 21, that the Easter Day is in 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	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	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. connection_timeout() - Deprecated from PHP 4.0.5. Checks whether the script has timed out. constant() - Returns the value of a constant. define() - Defines a constant.
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. attributes() - Returns the attributes/values of an element.
Stream	stream_copy_to_stream() - Copies data from one stream to another. stream_filter_append() - Appends a filter to a stream. stream_filter_prepend() stream_filter_register() stream_filter_remove()
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	<p>Below is a complete list of the timezones supported by PHP, which are useful with several PHP date functions.</p> <p>Africa America Antarctica Arctic Asia Atlantic Australia Europe Indian Pacific</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 commonly known as a regex (regexes) are a sequence of characters describing a special search pattern in the form of text string. They are basically used in programming world algorithms for matching some loosely defined patterns to achieve some relevant tasks. Sometimes regexes are understood as a mini programming language with a pattern notation which allows the users to parse text strings. The exact sequence of characters are unpredictable beforehand, so the regex helps in fetching the required strings based on a pattern definition.

As an example, you're given the task to check whether an e-mail has the correct form. Using a few commands, these problems can be solved thanks to regular expressions.

```
function validate_email($email_address)
{
    if( !preg_match("/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*@([a-zA-Z0-9_-])+([a-zA-Z0-9\._-]+)$/",$email_address))
    {
        return false;
    }
    return true;
}
```

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

```
<!DOCTYPE HTML>
<html>
<body>
    <form action="sample.php" method="get">
        Input here: <input type="text" name="value">
        <input type="submit">
    </form>
</body>
</html>
<?php
$Value = ($_GET['value']);
$pattern = "/^[^w]$Value/";

if (preg_match($pattern, 'The quick brown fox')){
    echo "$Value is found in the string";
}else{
    echo "$Value 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
$value = "$123,34.00A#";
echo preg_replace("/[^\0-9,.]/", "", $value);
?>
```

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, $match);
print $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: abcdeddf abcd der

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

Activity 4: Error Handling

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

- If the file does not exist.

Warning: fopen(mytestfile.txt): failed to open stream: No such file or directory in /wwwDeDKHB on line 6

Example code for handling:

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

Output:

Error: The file does not exist.

- A variable that does not exist.

Example code for handling:

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

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

//trigger error
echo($test);
?>
```

Output:

Error: [8] Undefined variable: test

- If a variable is bigger than the limit number (Example. "1")..

Example code for handling:

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

Output:

Notice: Value must be 1 or below in C:\xampp\htdocs\PHP\sample.php on line 4