

I ACTIVITY ANSWER SHEET

Name	DAN DAVID O. ABANILLA
Section:	BS-IT-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.

Reference: PHP Programming with MySQL, Don Gosselin 2006, USTP-CDO MAIN LIB

1. if	<i>If (conditional expression) statement</i>
2. if...else	<i>If (conditional expression) statement; else statement;</i>
3. if...else if...else	<i>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; }</i>
4. switch...case	<i>switch (expression){ case label: statement(s) break case label: statement(s) breaak; ... deafault: statement(s)</i>
5. for loop	<i>for (counter declaration and initialization; condition; update statement) { statement(s); }</i>
6. do while loop	<i>do { statement(s); }while (conditional expression);</i>
7. while loop	<i>while (conditional expression){ Statement(s); }</i>
8. foreach loop	<i>foreach (\$array_name as \$variable_name){ statements; }</i>
9. break statement	<i>\$array = array(values); foreach(\$array as \$value){ if(\$value == 3) break;</i>

10. continue statement	<pre>for (initialisation;condition;increment/decrement) { ... If (true condition) continue ... }</pre>
11. try...catch	<pre>try{ // code to be run } Catch(exception \$e){ // exception } Finally{ // optional code to run }</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

```
<?php
$p='1';
if (is_integer($p)){
    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
$p='0';
if ($p < 0){
    if ($p%2 ==0){
        echo "Negative and Even";
    }else {
        echo "Negative and Odd";
    }
} else {
    if ($p%2 == 0){
        echo "Positive and Even";
    }else
        echo "Positive 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
palindrome("dad");

function palindrome($drome) {
    $drome_len =strlen($drome) - 1;
    $result = "";

    for ($y = $drome_len; $y>=0; $y--) {
        $result .= $drome[$y];
    }
    if ($result == $drome){
        echo "Palindrome";
    }else {
        echo "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
$f=5;
$c=1;
for ($a=1; $a<=$f;$a++)
{
    $c=$c*$a;
}
echo "Factorial =".$c;
?>
```

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
$f=5;
$l=1;
$o=0;
$y=0;
for ($y= $f; $y> 0; $y--) {
    for ($o=$y; $o<$f; $o++){
        printf("%4o", $l);
        $l++;
    }

    echo nl2br("\n");
}

?>
```

Activity 2: PHP Built-in Functions

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

Array	array_change_key_cases <i>changes all keys in an array to uppercase</i> array_chunk <i>splits an array into chunks</i> array_column <i>retrieves the values from single column in an array</i> array_combine <i>merges the keys from one array and the values from another into a new array</i> array_count_values <i>counts all values in an array</i>
Calendar	cal_days_in_month() <i>Returns the number of days in a month for a specified year and calendar</i> cal_from_jd() <i>Converts a Julian Day Count into a date of a specified calendar</i> cal_info() <i>Returns information about a specified calendar</i> cal_to_jd() <i>Converts a date in a specified calendar to Julian Day Count</i> easter_date() <i>Returns the Unix timestamp for midnight on Easter of a specified year</i>
Date	checkdate() <i>Checks the validity of a Gregorian date</i> date_add() <i>Adds a number of days, months, years, hours, minutes and seconds to a date object</i> date_create_from_format() <i>Returns a formatted DateTime object</i> date_create() <i>Creates a new DateTime object</i> date_date_set() <i>Sets a new date</i>
Directory	chdir() <i>Changes the current directory</i> chroot() <i>Changes the root directory</i> closedir() <i>Closes a directory handle</i> dir() <i>Returns an instance of the Directory class</i> getcwd() <i>Returns the current working directory</i>
Error	debug_backtrace() <i>Used to generate a backtrace</i> debug_print_backtrace() <i>Prints a backtrace</i> error_get_last() <i>Gets the last error that occurred</i> error_log() <i>Sends an error message to the web server's log, a file or a mail account</i> error_reporting() <i>Specifies which PHP errors are reported</i>

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 if a variable of the specified type exists 25 of 38 filter_id() Returns the ID belonging to a named filter filter_input() Retrieves a specified external variable by name and optionally filters it filter_input_array() Pulls external variables and optionally filters them filter_list() Returns a list of all supported filters
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	to The recipient's email address. subject The email's subject line. message The actual email body where you can insert main messages. headers Additional parameters such as "From", "Cc", "Bcc" etc. Parameters Optional parameters.

Math	<code>abs()</code> <i>Returns the absolute (positive) value of a number</i> <code>acos()</code> <i>Returns the arc cosine of a number</i> <code>acosh()</code> <i>Returns the inverse hyperbolic cosine of a number</i> <code>asin()</code> <i>Returns the arc sine of a number</i> <code>asinh()</code> <i>Returns the inverse hyperbolic sine of a number</i>
Misc	<code>connection_aborted()</code> <i>Checks whether the client has disconnected</i> <code>connection_status()</code> <i>Returns the current connection status</i> <code>connection_timeout()</code> <i>Deprecated from PHP 4.0.5. Checks whether the script has timed out</i> <code>constant()</code> <i>Returns the value of a constant</i> <code>define()</code> <i>Defines a constant</i>
MySQLi	<code>mysqli_affected_rows()</code> <i>The number of affected rows in the previous MySQL operation</i> <code>mysqli_autocommit()</code> <i>Turn auto-committing database modifications on or off</i> <code>mysqli_change_user()</code> <i>Changes the user of the specified database connection</i> <code>mysqli_character_set_name()</code> <i>The default character set for the database connection</i> <code>mysqli_close()</code> <i>Closes an open database connection</i>
Network	<code>checkdnsrr()</code> <i>Checks DNS records for type corresponding to host</i> <code>closelog()</code> <i>Closes the connection of system logger</i> <code>define_syslog_variables()</code> <i>Deprecated and removed in PHP 5.4. Initializes the variables used in syslog functions</i> <code>dns_check_record()</code> <i>Alias of checkdnsrr()</i> <code>dns_get_mx()</code> <i>Alias of getmxrr()</i>
SimpleXML	<code>SimpleXMLElement::addAttribute</code> <i>Adds an attribute to the SimpleXML element</i> <code>SimpleXMLElement::addChild</code> <i>Adds a child element to the XML node</i> <code>SimpleXMLElement::asXML</code> <i>Return a well-formed XML string based on SimpleXML element</i> <code>SimpleXMLElement::attributes</code> <i>Identifies an element's attributes</i>

	SimpleXMLElement::children <i>Finds children of given node</i>
Stream	set_socket_blocking <i>Alias of stream_set_blocking</i> stream_bucket_append <i>Append bucket to brigade</i> stream_bucket_make_writeable <i>Return a bucket object from the brigade for operating on</i> stream_bucket_new <i>Create a new bucket for use on the current stream</i> stream_bucket_prepend <i>Prepend bucket to brigade</i>
String	addslashes() <i>Returns a string with backslashes in front of specified characters</i> addslashes() <i>Returns a string with backslashes in front of characters that need to be escaped</i> bin2hex() <i>Converts a string of ASCII characters to hexadecimal values</i> chop() <i>Removes space or other characters from the right end of a string</i> chr() <i>Returns a character from a specified ASCII value</i>
XML Parser	utf8_decode <i>Converts a string with ISO-8859-1 characters encoded with UTF-8 to single-byte ISO-8859-1</i> utf8_encode <i>Encodes an ISO-8859-1 string to UTF-8</i> xml_error_string <i>Get XML parser error string</i> xml_get_current_byte_index <i>Get current byte index for an XML parser</i> xml_get_current_column_number <i>Get current column number for an XML parser</i>
Zip	ZipArchive::addEmptyDir <i>Add a new directory</i> ZipArchive::addFile <i>Adds a file to a ZIP archive from the given path</i> ZipArchive::addFromString <i>Add a file to a ZIP archive using its contents</i> ZipArchive::addGlob <i>Add files from a directory by glob pattern</i> ZipArchive::addPattern <i>Add files from a directory by PCRE pattern</i>
Timezones	date_default_timezone_get() <i>Returns the default timezone used by all functions</i> date_default_timezone_set() <i>Sets the default timezone</i> Strtoftime()

	Short of time Gmdate() gm date
--	---

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. Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.
 - When checking that an email address entered into a form.

```
<?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
$string = "The quick brown fox";
$testing = "/Fox/i";
if (preg_match($testing, $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
$string = "The quick brown fox";
echo preg_replace('/\W\w+\s*(\W*)$/','',$string)."\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
$str = "$123,34.00A#";
echo preg_replace("/[^0-9.,]/", "", $str)."\n";
?>
```

- d. Write a PHP script to extract text (within parenthesis) from a string.

Sample String: 'The quick brown [fox].'
Expected output: Fox

```
<?php
$str = 'The quick brown [fox].';
preg_match('#\[([.*?])\]#', $str, $match);
print $match[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: abcdeddf d abcd der

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

Activity 4: Error Handling

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

*Parse Errors

```
try{
eval("echo 'toto' echo 'tata'");
```

```
}catch(ParseError $p){
```

```
    echo $p->getMessage();
}
```

*Fatal Errors

```
set_error_handler('myErrorHandler');
register_shutdown_function('fatalErrorShutdownHandler');
function myErrorHandler($code, $message, $file, $line) {
```

```
    ...
```

```
}
function fatalErrorShutdownHandler()
```

```
{
    $last_error = error_get_last();
    if ($last_error['type'] === E_ERROR) {
        // fatal error
        myErrorHandler(E_ERROR, $last_error['message'], $last_error['file'], $last_error['line']);
    }
}
```

*Warning Errors

```
set_error_handler("warning_handler", E_WARNING);
dns_get_record(...)
restore_error_handler();
```

```
function warning_handler($errno, $errstr) {
```

```
// do something

*Notice Errors
<?php

// Turn off all error reporting
error_reporting(0);

// Report simple running errors
error_reporting(E_ERROR | E_WARNING | E_PARSE);

// Reporting E_NOTICE can be good too (to report uninitialized
// variables or catch variable name misspellings ...)
error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);

// Report all errors except E_NOTICE
error_reporting(E_ALL & ~E_NOTICE);

// Report all PHP errors (see changelog)
error_reporting(E_ALL);

// Report all PHP errors
error_reporting(-1);

// Same as error_reporting(E_ALL);
ini_set('error_reporting', E_ALL);

?>
```