

## ACTIVITY ANSWER SHEET

Name	Leoner Vincent T. Cubillan
Section:	BSIT-3R2

**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>&lt;?php \$x = 11; if(\$x &gt; 10){     echo '\$x is greater than 10'; }</pre>
2. if...else	<pre>&lt;?php \$x = 2; if(\$x &gt; 10){     echo "\$x is greater than 10"; }else{     echo "\$x is less than 10"; } ?&gt;</pre>
3. if...else if...else	<pre>&lt;?php \$x = 20; if(\$x &gt; 0){     echo "\$x is greater than zero"; }elseif(\$x == 0){     echo "\$x is zero"; }else{     echo "\$x is less than zero"; } ?&gt;</pre>
4. switch...case	<pre>&lt;?php \$x = rand(0,3); switch(\$x){     case 0: echo 'x is equal 0';     break;     case 1: echo 'x is equal 1';     break;     case 2: echo 'x is equal 2';     break;     case 3: echo 'x is equal 3';     break; } ?&gt;</pre>
5. for loop	<pre>&lt;?php  for (\$i = 1; \$i &lt; 11; \$i++) {     echo \$i . '&lt;br/&gt;'; } ?&gt;</pre>
6. do while loop	<pre>&lt;?php  \$i = 10; do {     echo \$i . '&lt;br/&gt;';     \$i--; } while (\$i &gt; 0); ?&gt;</pre>

7. while loop	<pre>&lt;?php define('MAX',10);  \$counter = 1;  while(\$counter &lt;= MAX){     echo "The current value of the counter is {\$counter}&lt;br/&gt;";     \$counter++; } ?&gt;</pre>
8. foreach loop	<pre>foreach loop &lt;?php  \$scores = [1,2,3]; foreach (\$scores as \$score) {     echo \$score; } ?&gt;</pre>
9. break statement	<pre>&lt;?php  for(\$i=1;\$i&lt;=10;\$i++) {     echo "\$i &lt;br/&gt;";     if(\$i==5)     {         break;     } }  ?&gt;</pre>
10. continue statement	<pre>&lt;?php \$i= array(0=&gt;'Apple',1=&gt;'Orange','Banana', 'Mango'); foreach(\$i as \$v) {     if(\$v=='Mango')     {         continue;     }     echo "Value:". "\$v."."&lt;br&gt;."&lt;br&gt;"; } ?&gt;</pre>
11. try...catch	<pre>&lt;?php  function test() {     try {         throw new Exception('foo');     } catch (Exception \$e) {         return 'catch';     } finally {         return 'finally';     } }  echo test(); ?&gt;</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
$int = 1;
$int = '1';

if (filter_var($int, FILTER_VALIDATE_INT)) {
    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
function check($number){
    if($number % 2 == 0){
        echo "Positive and Even";
    }
    else{
        echo "Negative and Odd";
    }
}
$number = 39;
check($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

```
<?php
function Palindrome($string){
    if (strrev($string) == $string){
        return 1;
    }
    else{
        return 0;
    }
}

$original = "DAD";
if(Palindrome($original)){
    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
$n = 4;
$x = 1;
for($i=1;$i<=$n-1;$i++)
{
    $x*=( $i+1);
}
echo "$x"."\\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
echo "<pre>";
$num = 1;
for ($i = 1; $i <= 3; $i++) {
    for ($j = 1; $j <= $i; $j++) {
        echo $num . "&nbsp;";
        $num++;
        if ($j == $i) {
            echo "&nbsp;";
            echo "<br/>";
        }
    }
}
echo "</pre>";
?>
```

Activity 2: PHP Built-in Functions

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

Array	array() array_change_key_case() array_chunk() array_column() array_combine() array_count_values() array_diff() array_diff_assoc() array_diff_key() array_diff_uassoc() array_diff_ukey() array_fill() array_fill_keys() array_filter() array_flip() array_intersect() array_intersect_assoc() array_intersect_key() array_intersect_uassoc() array_intersect_ukey() array_key_exists() array_keys() array_map() array_merge() array_merge_recursive() array_multisort() array_pad() array_pop() array_product() array_push()
-------	---

	<div>array_rand() array_reduce() array_replace() array_replace_recursive() array_reverse() array_search() array_shift() array_slice() array_splice() array_sum() array_udiff() array_udiff_assoc() array_udiff_uassoc() array_uintersect() array_uintersect_assoc() array_uintersect_uassoc() array_unique() array_unshift() array_values() array_walk() array_walk_recursive() arsort() asort() compact() count() current() each() end() extract() in_array() key() krsort() ksort() list() natcasesort() natsort() next() pos() prev() range() reset() rsort() shuffle() sizeof() sort() uasort() uksort() usort()</div>
Calendar	<div>cal_days_in_month() cal_from_jd() cal_info() cal_to_jd() easter_date() easter_days() frenchtojd() gregoriantojd() jddayofweek() jdmonthname() jdtofrench() jdtogregorian() jdtojewish() jdtojulian()</div>

	<div>jdtounix() jewishtojd() juliantojd() unixtojd()</div>
Date	<div>checkdate() date_add() date_create_from_format() date_create() date_date_set() date_default_timezone_get() date_default_timezone_set() date_diff() date_format() date_get_last_errors() date_interval_create_from_date_string() date_interval_format() date_isodate_set() date_modify() date_offset_get() date_parse_from_format() date_parse() date_sub() date_sun_info() date_sunrise() date_sunset() date_time_set() date_timestamp_get() date_timestamp_set() date_timezone_get() date_timezone_set() date() getdate() gettimeofday() gmdate() gmmktime() gmstrftimee() idate() localtime() microtime() mktime() strftime() strtotime() strtotime() time() timezone_abbreviations_list() timezone_identifiers_list() timezone_location_get() timezone_name_from_abbr() timezone_name_get() timezone_offset_get() timezone_open() timezone_transitions_get() timezone_version_get()</div>
Directory	<div>chdir() chroot() closedir() dir() getcwd() opendir() readdir() rewinddir() scandir()</div>

Error	<div>debug_backtrace() debug_print_backtrace() error_get_last() error_log() error_reporting() restore_error_handler() restore_exception_handler() set_error_handler() set_exception_handler() trigger_error()</div>
File System	<div>basename() chgrp() chmod() chown() clearstatcache() copy() delete() dirname() disk_free_space() disk_total_space() diskfreespace() fclose() feof() fflush() fgetc() fgetcsv() fgets() fgetss() file() file_exists() file_get_contents() file_put_contents() fileatime() filectime() filegroup() fileinode() filemtime() fileowner() fileperms() filesize() filetype() flock() fnmatch() fopen() fpassthru() fputcsv() fputs() fread() fscanf() fseek() fstat() ftell() ftruncate() fwrite() glob() is_dir() is_executable() is_file() is_link() is_readable() is_uploaded_file()</div>

	<div>is_writable() is_writeable() lchgrp() lchown() link() linkinfo() lstat() mkdir() move_uploaded_file() parse_ini_file() parse_ini_string() pathinfo() pclose() popen() readfile() readlink() realpath() realpath_cache_get() realpath_cache_size() rename() rewind() rmdir() set_file_buffer() stat() symlink() tempnam() tmpfile() touch() umask() unlink()</div>
Filter	<div>filter_has_var() filter_id() filter_input() filter_input_array() filter_list() filter_var() filter_var_array()</div>
FTP	<div>ftp_alloc() ftp_cdup() ftp_chdir() ftp_chmod() ftp_close() ftp_connect() ftp_delete() ftp_exec() ftp_fget() ftp_fput() ftp_get() ftp_get_option() ftp_login() ftp_mdtm() ftp_mkdir() ftp_mlsd() ftp_nb_continue() ftp_nb_fget() ftp_nb_fput() ftp_nb_get() ftp_nb_put() ftp_nlist() ftp_pasv() ftp_put() ftp_pwd()</div>



	<div>ftp_quit() ftp_raw() ftp_rawlist() ftp_rename() ftp_rmdir() ftp_set_option() ftp_site() ftp_size() ftp_ssl_connect() ftp_systype()</div>
Libxml	<div>libxml_clear_errors() libxml_disable_entity_loader() libxml_get_errors() libxml_get_last_error() libxml_set_external_entity_loader() libxml_set_streams_context() libxml_use_internal_errors()</div>
Mail	<div>ezmlm_hash() mail()</div>
Math	<div>abs() acos() acosh() asin() asinh() atan() atan2() atanh() base_convert() bindec() ceil() cos() cosh() decbin() dechex() decoct() deg2rad() exp() expm1() floor() fmod() getrandmax() hexdec() hypot() intdiv() is_finite() is_infinite() is_nan() lcg_value() log() log10() log1p() max() min() mt_getrandmax() mt_rand() mt_srand() octdec() pi() pow() rad2deg() rand()</div>

	<div>round() sin() sinh() sqrt() srand() tan() tanh()</div>
Misc	<div>connection_aborted() connection_status() connection_timeout() constant() define() defined() die() eval() exit() get_browser() __halt_compiler() highlight_file() highlight_string() hrtime() ignore_user_abort() pack() php_strip_whitespace() show_source() sleep() sys_getloadavg() time_nanosleep() time_sleep_until() uniqid() unpack() usleep()</div>
MySQLi	<div>affected_rows autocommit change_user character_set_name close commit connect connect_errno connect_error data_seek debug dump_debug_info errno error error_list fetch_all fetch_array fetch_assoc fetch_field fetch_field_direct fetch_fields fetch_lengths fetch_object fetch_row field_count field_seek get_charset get_client_info get_client_stats get_client_version</div>

	<div>get_connection_stats</div> <div>get_host_info</div> <div>get_proto_info</div> <div>get_server_info</div> <div>get_server_version</div> <div>info</div> <div>init</div> <div>insert_id</div> <div>kill</div> <div>more_results</div> <div>multi_query</div> <div>next_result</div> <div>options</div> <div>ping</div> <div>poll</div> <div>prepare</div> <div>query</div> <div>real_connect</div> <div>real_escape_string</div> <div>real_query</div> <div>reap_async_query</div> <div>refresh</div> <div>rollback</div> <div>select_db</div> <div>set_charset</div> <div>set_local_infile_handler</div> <div>sqlstate</div> <div>ssl_set</div> <div>stat</div> <div>stmt_init</div> <div>thread_id</div> <div>thread_safe</div> <div>use_result</div> <div>warning_count</div>
Network	<div>checkdnsrr()</div> <div>closelog()</div> <div>dns_check_record()</div> <div>dns_get_mx()</div> <div>dns_get_record()</div> <div>fsockopen()</div> <div>gethostbyaddr()</div> <div>gethostbyname()</div> <div>gethostbynameel()</div> <div>gethostname()</div> <div>getmxrr()</div> <div>getprotobyname()</div> <div>getprotobynumber()</div> <div>getservbyname()</div> <div>getservbyport()</div> <div>header_register_callback()</div> <div>header_remove()</div> <div>header()</div> <div>headers_list()</div> <div>headers_sent()</div> <div>http_response_code()</div> <div>inet_ntop()</div> <div>inet_pton()</div> <div>ip2long()</div> <div>long2ip()</div> <div>openlog()</div> <div>pfssockopen()</div> <div>setcookie()</div>

	<div>setrawcookie() socket_get_status() socket_set_blocking() socket_set_timeout() syslog()</div>
SimpleXML	<div>__construct() __toString() addAttribute() addChild() asXML() attributes() children() count() getDocNamespaces() getName() getNamespaces() registerXPathNamespace() saveXML() simplexml_import_dom() simplexml_load_file() simplexml_load_string() xpath() current() getchildren() haschildren() key() next() rewind() valid()</div>
Stream	<div>set_socket_blocking() stream_bucket_prepend() stream_context_create() stream_context_get_default() stream_context_get_options() stream_context_get_params() stream_context_set_default() stream_context_set_options() stream_context_set_params() stream_copy_to_stream() stream_filter_append() stream_filter_prepend() stream_filter_register() stream_filter_remove() stream_get_contents() stream_get_filters() stream_get_line() stream_get_meta_data() stream_get_transports() stream_get_wrappers() stream_is_local() stream_isatty() stream_notification_callback() stream_register_wrapper()Alias of stream_wrapper_register() stream_resolve_include_path() stream_select() stream_set_blocking() stream_set_chunk_size() stream_set_read_buffer() stream_set_timeout() stream_set_write_buffer() stream_socket_accept()</div>

	<div>stream_socket_client() stream_socket_enable_crypto() stream_socket_get_name() stream_socket_pair() stream_socket_recvfrom() stream_socket_sendto() stream_socket_server() stream_socket_shutdown() stream_supports_lock() stream_wrapper_register() stream_wrapper_restore() stream_wrapper_unregister()</div>
String	<div>addslashes() addslashes() bin2hex() chop() chr() chunk_split() convert_cyr_string() convert_uudecode() convert_uuencode() count_chars() crc32() crypt() echo() explode() fprintf() get_html_translation_table() hebrex() hebrexc() hex2bin() html_entity_decode() htmlentities() htmlspecialchars_decode() htmlspecialchars() implode() join() lcfirst() levenshtein() localeconv() ltrim() md5() md5_file() metaphone() money_format() nl_langinfo() nl2br() number_format() ord() parse_str() print() printf() quoted_printable_decode() quoted_printable_encode() quotemeta() rtrim() setlocale() sha1() sha1_file() similar_text() soundex() sprintf()</div>

	<div>sscanf() str_getcsv() str_ireplace() str_pad() str_repeat() str_replace() str_rot13() str_shuffle() str_split() str_word_count() strcasecmp() strchr() strcmp() strcoll() strcspn() strip_tags() stripslashes() stripslashes() stripos() stristr() strlen() strnatcasecmp() strnatcmp() strncasecmp() strncmp() strpbrk() strpos() strrchr() strrev() stripos() strrpos() strspn() strstr() strtok() strtolower() strtoupper() strtr() substr() substr_compare() substr_count() substr_replace() trim() ucfirst() ucwords() vfprintf() vprintf() vsprintf() wordwrap()</div>
XML Parser	<div>utf8_decode() utf8_encode() xml_error_string() xml_get_current_byte_index() xml_get_current_column_number() xml_get_current_line_number() xml_get_error_code() xml_parse() xml_parse_into_struct() xml_parser_create_ns() xml_parser_create() xml_parser_free() xml_parser_get_option() xml_parser_set_option()</div>

	<div>xml_set_character_data_handler() xml_set_default_handler() xml_set_element_handler() xml_set_end_namespace_decl_handler() xml_set_external_entity_ref_handler() xml_set_notation_decl_handler() xml_set_object() xml_set_processing_instruction_handler() xml_set_start_namespace_decl_handler() xml_set_unparsed_entity_decl_handler()</div>
Zip	<div>zip_close() zip_entry_close() zip_entry_compressedsize() zip_entry_compressionmethod() zip_entry_filesize() zip_entry_name() zip_entry_open() zip_entry_read() zip_open() zip_read()</div>
Timezones	<div>Africa America Antarctica Arctic Asia Atlantic Australia Europe Indian Pacific</div>

### Activity 3: Regular Expression

1. Define Regular Expression (RegEx) and provide example programming scenario where you can use (RegEx). Provide example syntax in PHP.

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
$word = "fox";
$string = "The quick brown fox";

// Test if string contains the word
if(strpos($string, $word) !== false){
    echo "Fox is found the string";
} else{
    echo "Fox Not Found!";
}
?>
```

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
$str1 = 'The quick brown fox';
echo preg_replace('/\W\w+\s*(\W*)$/ ', '$1', $str1)."\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
$str1 = "$12,334.00A";
echo preg_replace("/[^0-9,.]/", "", $str1)."\n";
?>
```

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

Sample String: 'The quick brown [fox].'

Expected output: Fox

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

## Activity 4: Error Handling

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

```
<?php
$test = 5;
if ($test > 1) {
    trigger_error('Value of $test must be 1 or less', E_USER_NOTICE);
}
```

**The example below records errors to a database table database whenever one is encountered**

```
<?php
function errorHandler($errno, $errstr, $errfile, $errline) {
    static $db;
    if (empty($db)) {
        $db = new PDO(DSN, DBUSER, DBPASS);
    }

    $query = "INSERT INTO errorlog (severity, message, filename, lineno, time) VALUES
    (?, ?, ?, ?, NOW())";
    $stmt = $db->prepare($query);

    switch ($errno) {
        case E_NOTICE:
        case E_USER_NOTICE:
        case E_DEPRECATED:
        case E_USER_DEPRECATED:
        case E_STRICT:
            $stmt->execute(array("NOTICE", $errstr, $errfile, $errline));
            break;

        case E_WARNING:
        case E_USER_WARNING:
            $stmt->execute(array("WARNING", $errstr, $errfile, $errline));
            break;

        case E_ERROR:
        case E_USER_ERROR:
            $stmt->execute(array("FATAL", $errstr, $errfile, $errline));
            exit("FATAL error $errstr at $errfile:$errline");
    }
}
```

```

        default:
            exit("Unknown error at $errfile:$errline");
    }
}

set_error_handler("errorHandler");

$test = 5;
if ($test > 1) {
    trigger_error("Value of $test must be 1 or less", E_USER_NOTICE);
}

```

## Handling Errors using Exceptions

```

<?php
try {
    $data = $this->getDataFromService();
}
catch (Exception $e) {
    echo "Caught exception: " . $e->getMessage() . "n";
}

```