

ACTIVITY ANSWER SHEET

Name	Mark Vincent R. Achacoso
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)</pre>

	<pre>{ code to be executed for each iteration; }</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	<pre>break;</pre>
10. continue statement	<pre>continue;</pre>
11. try...catch	<pre>try { // run your code here } catch (exception \$e) { //code to handle the exception } finally { //optional code that always runs }</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

$myNumber = "1";

if(is_int($myNumber)){
    echo "$myNumber is a number";
}
else {
    echo "$myNumber is not a number";
}

?>
```

b. Write a program that checks if a value is positive or negative and odd or even.

Sample input: 0

Expected output: Positive & Even

Sample input: -1

Expected output: Negative and Odd

```
<?php

$myNumber = 1;

function OddOrEven(){
    global $myNumber;

    if($myNumber%2==0){
        echo "Even";
    }
    else{
        echo "Odd";
    }
}

function PositiveOrNegative(){
    global $myNumber;

    if($myNumber >= 0){
        echo "Positive";
    }
    else{
        echo "Negative";
    }
}

OddOrEven();
echo " and ";
PositiveOrNegative();

?>
```

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

$myString = "anna";
#$myString = "bogart";
```

```

    if($myString == strrev($myString)){
        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

    $myNumber = 4;
    $myFactorial = 1;

    for ($i = 1; $i <= $myNumber; $i++){
        $myFactorial = $myFactorial * $i;
    }
    echo "$myFactorial"

?>

```

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 ($i = 1; $i <= $myNumber; $i++){
        for ($j = 1; $j <= $i; $j++){
            echo $count . "&nbsp;";
            $count++;
            if ($j == $i){

```

```
        echo "&nbsp;";
        echo "<br/>";
    }
}
}

?>
```

Activity 2: PHP Built-in Functions

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

Array	<p>The <code>array()</code> function is used to create an array.</p> <p>The <code>array_change_key_case()</code> function changes all keys in an array to lowercase or uppercase.</p> <p>The <code>array_chunk()</code> function splits an array into chunks of new arrays.</p> <p>The <code>array_column()</code> function returns the values from a single column in the input array.</p>
Calendar	<p><code>cal_days_in_month(<i>calendar</i>,<i>month</i>,<i>year</i>);</code></p> <p><code>cal_from_jd(<i>jd</i>,<i>calendar</i>);</code></p> <p><code>cal_info(<i>calendar</i>);</code></p> <p><code>cal_to_jd(<i>calendar</i>,<i>month</i>,<i>day</i>,<i>year</i>);</code></p>
Date	<p>The <code>checkdate()</code> function is used to validate a Gregorian date.</p>

	<p>The <code>date_add()</code> function adds some days, months, years, hours, minutes, and seconds to a date.</p> <p>The <code>date_create_from_format()</code> function returns a new <code>DateTime</code> object formatted according to the specified format.</p> <p>The <code>date_create()</code> function returns a new <code>DateTime</code> object.</p> <p>The <code>date_date_set()</code> function sets a new date.</p>
Directory	<p>The <code>chdir()</code> function changes the current directory.</p> <p>The <code>chroot()</code> function changes the root directory of the current process to <i>directory</i>, and changes the current working directory to <code>"/"</code>.</p> <p>The <code>closedir()</code> function closes a directory handle.</p> <p>The <code>dir()</code> function returns an instance of the <code>Directory</code> class.</p> <p>The <code>getcwd()</code> function returns the current working directory.</p>
Error	<p>The <code>debug_backtrace()</code> function generates a PHP backtrace.</p> <p>The <code>debug_print_backtrace()</code> function prints a PHP backtrace.</p> <p>The <code>error_get_last()</code> function returns the last error that occurred (as an associative array).</p> <p>The <code>error_log()</code> function sends an error message to a log, to a file, or to a mail account.</p>
File System	<p>The <code>basename()</code> function returns the filename from a path.</p> <p>The <code>chgrp()</code> function changes the usergroup of the specified file.</p> <p>The <code>chmod()</code> function changes permissions of the specified file.</p> <p>The <code>chown()</code> function changes the owner of the specified file.</p> <p>The <code>clearstatcache()</code> function clears the file status cache.</p> <p>The <code>copy()</code> function copies a file.</p>
Filter	<p>The <code>filter_has_var()</code> function checks whether a variable of a specified input type exist.</p>

	<p>The <code>filter_id()</code> function returns filter ID of a specified filter name.</p> <p>The <code>filter_input()</code> function gets an external variable (e.g. from form input) and optionally filters it.</p> <p>The <code>filter_input_array()</code> function gets external variables (e.g. from form input) and optionally filters them.</p>
FTP	<p>The <code>ftp_alloc()</code> function allocates space for a file to be uploaded to the FTP server.</p> <p>The <code>ftp_cdup()</code> function changes to the parent directory on the FTP server.</p> <p>The <code>ftp_chdir()</code> function changes the current directory on the FTP server.</p> <p>The <code>ftp_chmod()</code> function sets permissions on the specified file via FTP.</p> <p>The <code>ftp_close()</code> function closes an FTP connection.</p>
Libxml	<p>The <code>libxml_clear_errors()</code> function clears the libxml error buffer.</p> <p>The <code>libxml_disable_entity_loader()</code> function enables the ability to load external entities.</p> <p>The <code>libxml_get_errors()</code> function gets the errors from the the libxml error buffer.</p> <p>The <code>libxml_get_last_error()</code> function gets the last error from the libxml error buffer.</p>
Mail	<p>The <code>mail()</code> function allows you to send emails directly from a script.</p> <p>The <code>ezmlm_hash()</code> function calculates the hash value needed when keeping EZMLM mailing lists in a MySQL database.</p>
Math	<p>The <code>abs()</code> function returns the absolute (positive) value of a number.</p> <p>The <code>acos()</code> function returns the arc cosine of a number.</p> <p>The <code>acosh()</code> function returns the inverse hyperbolic cosine of a number.</p> <p>The <code>asin()</code> function returns the arc sine of a number.</p>
Misc	<p>The <code>connection_aborted()</code> function checks whether the client has disconnected.</p> <p>The <code>connection_timeout()</code> function checks whether the script has timed out.</p> <p>The <code>constant()</code> function returns the value of a constant.</p>

	The define() function defines a constant.
MySQLi	<p>The affected_rows / mysqli_affected_rows() function returns the number of affected rows in the previous SELECT, INSERT, UPDATE, REPLACE, or DELETE query.</p> <p>The autocommit() / mysqli_autocommit() function turns on or off auto-committing database modifications.</p> <p>The close() / mysqli_close() function closes a previously opened database connection.</p> <p>The connect() / mysqli_connect() function opens a new connection to the MySQL server.</p>
Network	<p>The checkdnsrr() function checks DNS records for <i>type</i> corresponding to <i>host</i>.</p> <p>The closelog() function closes the connection of system logger.</p> <p>The dns_check_record() function is an alias of the checkdnsrr() function.</p>
SimpleXML	<p>The __construct() function creates a new SimpleXMLElement object.</p> <p>The __toString() function returns the string content of an element.</p> <p>The addAttribute() function appends an attribute to the SimpleXML element.</p>
Stream	<p>stream_context_get_params()</p> <p>stream_bucket_prepend()</p> <p>stream_context_get_default()</p> <p>stream_context_set_options()</p>
String	<p>The addcslashes() function returns a string with backslashes in front of the specified characters.</p> <p>The addslashes() function returns a string with backslashes in front of predefined characters.</p> <p>The chop() function removes whitespaces or other predefined characters from the right end of a string.</p>

XML Parser	<p>The utf8_decode() function decodes a UTF-8 string to ISO-8859-1.</p> <p>The utf8_encode() function encodes an ISO-8859-1 string to UTF-8.</p> <p>The xml_error_string() function returns the XML parser error description.</p>
Zip	<p>The zip_close() function closes a ZIP file archive opened by the zip_open() function.</p> <p>The zip_entry_close() function closes a ZIP directory entry opened by zip_entry_open().</p> <p>The zip_entry_compressionmethod() function returns the compression method of a ZIP directory entry.</p>
Timezones	<p>PHP date_default_timezone_set() Function</p> <p>PHP IntlChar getBidiPairedBracket() Function</p> <p>PHP date_get_last_errors() Function</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.

```
<?php
```

```
$myString = "My name is mark";
```

```
if (preg_match("/mark/", $myString)){
```

```
    echo "mark is found the string";
```

```
}
```

```
else{
```

```
    echo "mark is not found the string";
```

```
}
```

```
?>
```

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
$myString = "The quick brown fox";

if (preg_match("/fox/", $myString)){
    echo "Fox is found the string";
}
else{
    echo "Fox is not found 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
$myString = "The quick brown fox";
$myRemoveString = preg_replace('/\W\w+\s*(\W*)$/', '$1', $myString);

echo $myRemoveString
?>
```

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

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

Expected output: 123,34.00

```
<?php
$myString = "$123,34.00A#";
$removeString = preg_replace("/[^0-9,.]/", "", $myString);
echo $removeString
?>
```

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

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

Expected output: Fox

```
<?php
$myString = "The quick brown [Fox].";
preg_match('#\[(.*?)\]#', $myString, $myMatch);
print $myMatch[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
$myAlphabet = 'abcde$ddfd @abcd )der]';
$myOutput = preg_replace("/[^A-Za-z0-9 ]/", '', $myAlphabet);
echo $myOutput
?>
```

Activity 4: Error Handling

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

```
<?php

// Set MySQLi to throw exceptions
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
try {
    $connection2 = mysqli_connect('localhost', 'root1', 'Abc123456!', 'testx');
} catch (mysqli_sql_exception $ex) {
    throw new Exception("Can't connect to the database! \n" . $ex);
}
```

?>

Fatal error: Uncaught Exception: Can't connect to the database! mysqli_sql_exception: Access denied for user 'root1'@'localhost' (using password: YES) in C:\xampp\htdocs\act.php:6 Stack trace: #0 C:\xampp\htdocs\act.php(6): mysqli_connect('localhost', 'root1', 'Abc123456!', 'testx') #1 {main} in C:\xampp\htdocs\act.php:8 Stack trace: #0 {main} thrown in C:\xampp\htdocs\act.php on line 8

Using die() or exit() for handling errors

```
<?php

mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
try {
    $connection2 = mysqli_connect('localhost', 'root1', 'Abc123456!', 'testx');
} catch (mysqli_sql_exception $ex) {
    die("Can't connect to the database! \n" . $ex);
}
```

?>