ACTIVITY ANSWER SHEET

Name	Kristine C.Dumdum
Section:	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	`if (condition) {`
	`  code to be executed if condition is true;`
	`}`

2. if…else	`if (condition) {`
	`  code to be executed if condition is true;`
	`} else {`
	`  code to be executed if condition is false;`
	`}`

3. if…else if…else	`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;`
	`}`

4. switch…case	`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;`
	`}`

5. for loop	`for (init counter; test counter; increment counter) {`
	`  code to be executed;`
	`}`

6. do while loop	`do {`
	`  code to be executed;`
	`} while (condition is true);`

7. while loop	`while (condition is true) {`
	`  code to be executed;`
	`}`

8. foreach loop	`foreach ($array as $value) {`
	`  code to be executed;`
	`}`

9. break statement	`$i = 0;`
	
	`while( $i < 10) {`
	`  $i++;`
	`  if( $i == 3 )break;`
	`}`
	`echo ("Loop stopped at i = $i" );`

10. continue statement	`$array = array( 1, 2, 3, 4, 5);` `foreach( $array as $value ) {` `if( $value == 3 )continue;` `echo "Value is $value  ";` `}`

11. try…catch	`try {`
	`$error = 'Always throw this error';`
	`throw new Exception($error);`
	
	`// Code following an exception is not executed.`
	`echo 'Never executed';`
	`}catch (Exception $e) {`
	`echo 'Caught exception: ', $e->getMessage(), "\n";`
	`}`
	
	`// Continue execution`
	`echo 'Hello World';`

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

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

    $UserInput = ($_GET['UserInput']);

    if(is_numeric($UserInput)){
        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

```
<!DOCTYPE html>
<html>
<body>
<form action="index.php" method="get">
        Input: <input type="number" name="UserInput">
        <input type="submit">
    </form>
</body>
</html>
<?php
    $UserInput = ($_GET['UserInput']);

    if ($UserInput >= 0) {
        if ($UserInput % 2 == 0) {
            echo "Positive & Even";
        }else{
            echo "Positive & Odd";
        }
    }else {
        if ($UserInput % 2 == 0) {
            echo "Negative & Even";
        }else{
            echo "Negative & Odd";
        }
    }
?>
```

c. Write a program that checks if a value is palindrome.

Sample input: Anna
Expected output: Palindrome

Sample input: Bogart
Expected output: Not a Palindrome

```
<!DOCTYPE html>
<html>
<body>
<form action="index.php" method="get">
        Input: <input type="text" name="UserInput">
        <input type="submit">
    </form>
</body>
</html>
<?php
    $UserInput = ($_GET['UserInput']);
    if (strtolower($UserInput) == strrev(strtolower($UserInput))) {
        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="index.php" method="get">
        Input: <input type="text" name="UserInput">
        <input type="submit">
    </form>
</body>
</html>
<?php
    $UserInput = ($_GET['UserInput']);
    $i = 1; $factorial=1;
    for ($i=1; $i<=$UserInput; $i++){
        $factorial = $i*$factorial;
    }
    echo “Factorial: $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="index.php" method="get">
        Input: <input type="text" name="UserInput">
        <input type="submit">
    </form>
</body>
</html>
<?php
    $UserInput = ($_GET['UserInput']);
    $a = 1; $b=1; $count=1;
    for ($a=1; $a<=$UserInput; $a++) {
        for ($b=1; $b<=$a; $b++) {
            echo "$count     ";
            $count++;
            if ($a == $b) {
                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_combine() Creates an array by using the elements from one "keys" array and one "values" 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_date()Returns the Unix timestamp for midnight on Easter of 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	error_get_last() function returns the last error that occurred (as an associative array) error_log() function sends an error message to a log, to a file, or to a mail account error_reporting() function specifies which errors are reported restore_error_handler() function restores the previous error handler set_error_handler() function sets a user-defined error handler function
File System	basename() function returns the filename from a path chgrp() function changes the usergroup of the specified file chmod() function changes permissions of the specified file chown() function changes the owner of the specified file clearstatcache() function 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	ibxml_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 constant() Returns the value of a constant define() Defines a constant defined() Checks whether a constant exists
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	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 fsockopen() Opens an Internet or Unix domain socket connection gethostbyaddr() Returns the domain name for a given IP address
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 asXML() Returns a well-formed XML string (XML version 1.0) from a SimpleXML object
Stream	stream_bucket_prepend() stream_context_create() stream_context_get_default() stream_context_get_options() stream_context_get_params()
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	timezone_abbreviations_list() Returns an associative array containing dst, offset, and the timezone name timezone_identifiers_list() Returns an indexed array with all timezone identifiers timezone_location_get() Returns location information for a specified timezone timezone_name_from_ abbr()Returns the timezone name from abbreviation timezone_name_get() Returns the name of the timezone

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.

PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression.

Regular expressions are powerful pattern matching algorithm that can be performed in a single expression.

Regular expressions use arithmetic operators such as (+,-,^) to create complex expressions.

Regular expressions help you accomplish tasks such as validating email addresses, IP address etc.

Basic Syntax:

```
<?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

```
<!DOCTYPE html>
<html>
<body>
<form action="index.php" method="get">
        Input: <input type="text" name="UserInput">
        <input type="submit">
    </form>
</body>
</html>
<?php
    $UserInput = ($_GET['UserInput']);
    $strings = "the quick brown fox";
    if (preg_match(("/$UserInput/"), $strings)) {
        echo "$UserInput is found in the string";
    }else {
        echo "$UserInput 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
    $myString = "The quick brown fox";
    $removeLast = preg_replace('/\W\w+\s*(\W*)$/', '$1', $myString);
    echo "$removeLast"
?>
```

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#";
    $removed = preg_replace('/[^0-9,.]/', '', $myString);
    echo "$removed"
?>
```

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, $parenthesis);
    print $parenthesis[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: abcdeddfd abcd der

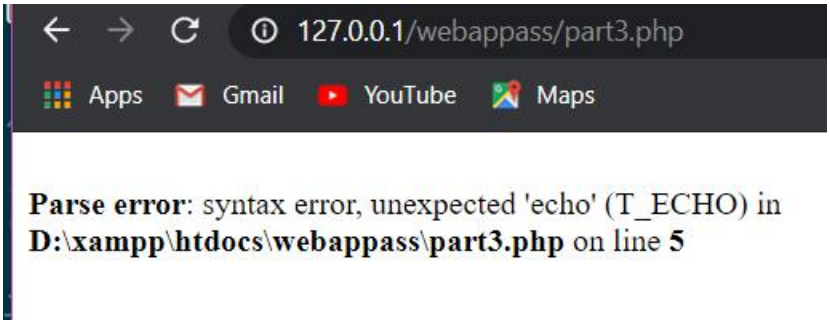
```
<?php
    $myString = 'abcde$ddfd @abcd )der]';
    echo 'Old string : '.$myString.'<br>';
    $replace = preg_replace("/[^A-Za-z0-9 ]/", '', $myString);
    echo 'New string : '.$replace."<br>";
?>
```

Activity 4: Error Handling

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

1. **Parse error or Syntax Error:** It is the type of error done by the programmer in the source code of the program. The syntax error is caught by the compiler. After fixing the syntax error the compiler compile the code and execute it. Parse errors can be caused dues to unclosed quotes, missing or Extra parentheses, Unclosed braces, Missing semicolon etc.

To solve the missing parenthesis error in PHP, the code has to be checked from the beginning to search for it. One way to avoid errors is to use proper indentation in the code.



2. **Fatal Error:** It is the type of error where PHP compiler understand the PHP code but it recognizes an undeclared function. This means that function is called without the definition of function.

To avoid the error from the example above, we need to create the proper code to handle an exception.