ACTIVITY ANSWER SHEET

Name	Acenas Jovane Rey G.
Section:	3R1

Instructions:

1. Push your output on your GITHUBrepository.
2. Use the answer sheet provided saveit 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 the condition is true;
	}

2. if…else	If (condition) {
	Code to be executed if the condition if true;
	} else
	Code to be executed if the condition is if false;
	}

3. if…else if…else	If (condition) {
	Code to be executed if this condition if true;
	} elseif (condition)
	Code to be executed if this condition is false and this condition is true;
	}else
	{ Code to be executed if all condition 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 for each iteration;`
	`}`

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	`for (i = 0; i < 10; i++) {` `  if (i === 3) { break; }` `  text += "The number is " + i + " ";` `}`
10. continue statement	`for (i = 0; i < 10; i++) {` `  if (i === 3) { continue; }` `  text += "The number is " + i + " ";` `}`

11. try…catch	`try {`
	`  //  Block of code to try`
	`}`
	`catch(Exception e) {`
	`  //  Block of code to handle errors`
	`}`

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

```
<form action="checkifnumber.php" method="post">
    Value: <input type="text" name="Value" onClick='clearform();'/>
</form>

<?php
    if (isset($_POST['Value'])){
        $Value = ($_POST['Value']);
    }
    if (filter_var($Value, FILTER_VALIDATE_INT)===0||!filter_var($Value, FILTER
_VALIDATE_INT)===FALSE){
       echo "A number<br>";
    }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

```
<form action="oddeven.php" method="post">
    Value: <input type="text" name="Value" onClick='clearform();'/>
</form>

<?php
 $num=$_POST['Value'];{
 $Value = ($_POST['Value']);
 if($Value%2==0)
 {
   if($Value>0){
    echo $Value." is positive even number";
```

```
   }
   else{
    echo $Value." is negative even number";
   }
 }

 else

 {
  if($Value>0){
    echo $Value." is positive odd number";
  }
  else{
    echo $Value." is negative odd 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

```
<form action="palindrome.php" method="post">
    Value: <input type="text" name="Value" onClick='clearform();'/>
</form>

<?php
$string=$_POST['Value'];{
$Value = ($_POST['Value']);
function Palindrome($Value){
    if (strrev($Value) == $Value){
        return 1;
    }
    else{
        return 0;
    }
}

if(Palindrome($Value)){
    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

```
<form action="fial.php" method="post">
    Value: <input type="text" name="Value" onClick='clearform();'/>
</form>

<?php
    if (isset($_POST['Value'])){
    $Value = ($_POST['Value']);
$x = 1;
for($i=1;$i<=$Value-1;$i++)
{
 $x*=($i+1);
}
echo "The factorial of  $Value = $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
```

```
<form action="floyd.php" method="post">
    Value: <input type="text" name="Value" onClick='clearform();'/>
</form>

<?php
    if (isset($_POST['Value'])){
    $Value = ($_POST['Value']);

 echo "Value = " . $Value . "\n <br>";
 $count = 1;
 for ($i = $Value; $i > 0; $i--)
 {
   for ($j = $i; $j < $Value + 1; $j++)
    {
      printf("%4s", $count);
      $count++;
    }
     echo "\n <br>";
    }
    }
 ?>
```

Activity 2: PHP Built-in Functions

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

Array	The array functions allow you to access and manipulate arrays.
Calendar	The calendar extension contains functions that simplifies converting between different calendar formats.
Date	The date/time functions allow you to get the date and time from the server where your PHP script runs. You can then use the date/time functions to format the date and time in several ways.
Directory	The directory functions allow you to retrieve information about directories and their contents.
Error	The error functions are used to deal with error handling and logging. The error functions allow us to define own error handling rules, and modify the way the errors can be logged. The logging functions allow us to send messages directly to other machines, emails, or system logs. The error reporting functions allow us to customize what level and kind of error feedback is given.
File System	The filesystem functions allow you to access and manipulate the filesystem.
Filter	This PHP filters is used to validate and filter data coming from insecure sources, like user input.
FTP	The FTP functions give client access to file servers through the File Transfer Protocol (FTP). The FTP functions are used to open, login and close connections, as well as upload, download, rename, delete, and get information on files from file servers. Not all of the FTP functions will work with every server or return the same results. The FTP functions became available with PHP 3.

Libxml	The libxml functions and constants are used together with SimpleXML, XSLT and DOM functions.
Mail	The mail() function allows you to send emails directly from a script.
Math	The math functions can handle values within the range of integer and float types.
Misc	The misc. functions were only placed here because none of the other categories seemed to fit.
MySQLi	The MySQLi functions allows you to access MySQL database servers.
Network	The Network functions contains various network function and let you manipulate information sent to the browser by the Web server, before any other output has been sent.
SimpleXML	SimpleXML is an extension that allows us to easily manipulate and get XML data. SimpleXML provides an easy way of getting an element's name, attributes and textual content if you know the XML document's structure or layout. SimpleXML turns an XML document into a data structure you can iterate through like a collection of arrays and objects.
Stream	The Stream functions .... Streams are the way of generalizing file, network, data compression, and other operations which share a common set of functions and uses. In its simplest definition, a stream is a resource object which exhibits streamable behavior. That is, it can be read from or written to in a linear fashion, and may be able to fseek() to an arbitrary location within the stream.
String	The PHP string functions are part of the PHP core. No installation is required to use these functions.

XML Parser	he XML functions lets you parse, but not validate, XML documents. XML is a data format for standardized structured document exchange. More information on XML can be found in our XML Tutorial. This extension uses the Expat XML parser.
Zip	The Zip files functions allows you to read ZIP files.
Timezones	Useful with several PHP date functions.

Activity 3: Regular Expression

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

Answer: Regular expressions are specially encoded text strings used as patterns for matching sets of strings. They began to emerge in the 1940s as a way to describe regular languages, but they really began to show up in the programming world during the 1970s. The first place I could find them showing up was in the QED text editor written by Ken Thompson.

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
$pattern = '/[^\w]fox\s/';
if (preg_match($pattern, 'The quick brown fox jumps over the lazy
dog'))
  {
  echo "'fox' is present..."."\n";
  }
  else
  echo "'fox' is not present..."."\n";
?>
```

b. Write a PHP script that removes the last word from a string.
Sample String: 'The quick brown fox'
Expected output: 'The quick brown'

```
<form action="last.php" methode=post">
Value: <input type="text" name="Value" onClick='clearform();'/>

<?php
If (isset($_POST['Value'])){
$Value = ($_POST['Value']);

echo preg_replace('/\W\w+\s*(\W*)$/', '$1', $Value)."\n";
}
?>
```

c. Write a PHP script to remove nonnumeric characters except comma and dot.
Sample String:'/$123,34.00A#'
Expected output:123,34.00

```
<form action="last.php" methode=post">
Value: <input type="text" name="Value" onClick='clearform();'/>

<?php
If (isset($_POST['Value'])){
$Value = ($_POST['Value']);
```

```
echo preg_replace("/[^0-9,.]/", "", $Value)."\n";
}
?>
```

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

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

Expected output: Fox

```
<form action="last.php" methode=post">
Value: <input type="text" name="Value" onClick='clearform();'/>

<?php
If (isset($_POST['Value'])){
$Value = ($_POST['Value']);

echo preg_replace('/\s+/', ' ', trim($Value))."\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:abcdeddfdabcd 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.