ACTIVITY ANSWER SHEET

Name	SETH P. BAJAO
Section:	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	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 first condition is false and this condition is true; } else { code to be executed if all conditions are false; }
4. switch…case	switch (*n*) { case n*label1:* *code to be executed if n=label1;* break; case *label2:* *code to be executed if n=label2;* 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	jump statement; break;
10. continue statement	Loop (While, do-while, for,) { conditions { continue;//continue statement } code executed; }
11. try…catch	try { // run your code here } catch (exception $e) { //code to handle the exception } finally { //optional code that always runs }

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
if ( (int) '1'!== 1 ) {
	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 "Even ";
	}
	else{
		echo "Odd ";
	}
}
function sample($number){
	if($number >= 0){
		echo "&Positive";
	}
	else{
		echo "&Negative";
	}
}
$number = -1;
check($number);
sample($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;
	}
}

// Driver Code
$original = "anna";
if(Palindrome($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 "Factorial of $n is = $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
$n = 3;
echo "sample input = ". $n . "\n";
$count = 1;
for ($i = $n; $i >0; $i--)
{
    for ($j = $i; $j <$n + 1; $j++)
     {
        printf("%4s", $count);
        $count++;
     }
          echo "\n";
     }
?>
```

Activity 2: PHP Built-in Functions

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

Array	array_count_values() array_diff() array_diff_assoc() array_diff_key() array_diff_uassoc()
Calendar	cal_days_in_month() cal_from_jd() cal_info() cal_to_jd() easter_date()
Date	date_create_from_format() date_create() date_date_set() date_default_timezone_get() date_default_timezone_set()
Directory	chdir() chroot() closedir() dir() getcwd()
Error	debug_backtrace() debug_print_backtrace() error_clear_last() error_get_last() error_log()
File System	basename() chgrp() chmod() chown() clearstatcache()
Filter	filter_has_var() filter_id() filter_input() filter_input_array() filter_list()
FTP	ftp_get_option() ftp_login() ftp_mdtm() ftp_mkdir() ftp_mlsd()
Libxml	libxml_clear_errors() libxml_disable_entity_loader() libxml_get_errors() libxml_get_last_error()

	libxml_set_external_entity_loader()
Mail	ezmlm_hash() mail()
Math	intdiv() is_finite() is_infinite() is_nan()' lcg_value()
Misc	define() defined() die() eval() exit()
MySQLi	fetch_assoc() fetch_field() fetch_field_direct() fetch_fields() fetch_lengths()
Network	gethostbyaddr() gethostbyname() gethostbynamel() gethostname() getmxrr()
SimpleXML	addChild() asXML() attributes() children() count()
Stream	stream_get_meta_data() stream_get_transports() stream_get_wrappers() stream_is_local() stream_isatty()
String	htmlspecialchars_decode() htmlspecialchars() implode() join() lcfirst()
XML Parser	xml_parse_into_struct() xml_parser_create_ns() xml_parser_create() xml_parser_free() xml_parser_get_option

Zip	zip_entry_open() zip_lesize() entry_read() zip_open() zip_read()
Timezones	Africa America Antarctica Arctic Asia

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 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.

```
<?php

$my_text="I Love Regular Expressions";

$my_array    = preg_split("/ /", $my_text);

print_r($my_array );

?>
```

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
$str1 = 'The quick brown fox.';
if (strpos($str1,'fox') !== false)
  {
      echo 'Fox is found the string.';
  }
else
  {
      echo 'Fox is not found as 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
$str1 = 'The quick brown fox';
echo preg_replace('/\W\w+\s*(\W*)$/', '$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 = "/$123,34.00A#";
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: abcdeddfd abcd der

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

Activity 4: Error Handling

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

A. **Die statements**– the die function combines the echo and exit function in one. It is very useful when we want to output a message and stop the script execution when an error occurs.

```
<?php
$site = "https://SETHBAJAO.COM";
fopen($site,"r")
or die("Unable to connect to $site");
?>
```

B. **Custom error handlers** – these are user defined functions that are called whenever an error occurs.

error_function(error_level,error_message,
error_file,error_line,error_context)

C. **PHP error reporting** – the error message depending on your PHP error reporting settings. This method is very useful in development environment when you have no idea what caused the error. The information displayed can help you debug your application.

```
function customError($errno, $errstr) {
   echo "<b>Error:</b> [$errno] $errstr<br>";
   echo "Ending Script";
   die();
}
```