ACTIVITY ANSWER SHEET

Name	Geraldine P. Aruta
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 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 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 | ```
break;
``` |
| 10. continue statement | For (initialization;condition;increment/decrement)
  {
    …
     If (True Condition)

        Continue;

     …
    } |
| 11. try…catch | ```
function checkNum($number) {
  if($number>1) {
    throw new Exception("Value must be 1 or below");
  }
  return true;
}

//trigger exception in a "try" block
try {
  checkNum(2);
  //If the exception is thrown, this text will not be shown
  echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
  echo 'Message: ' .$e->getMessage();
}
``` |

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
<?php
$x = 3;
$y = '3';
if (is_int($x)) {
    echo "$x is integer\n";
} else {
    echo "$x is not an integer";
}
echo "<br>";

if (is_int($y)) {
    echo "$y is integer\n";
```

```
} else {
    echo "$y is not an integer <br> ";
}

?>
```

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
<?php
$x = 3;
if ($x >= 0){
        if ($x % 3 == 0){
        echo "$x is Positive and Even";
    echo "<br>";
  }
  else{
        echo "$xr is Positive and Odd";
    echo "<br>";
  }
}
else{
        if ($xr % 3 == 0){
        echo "$xr is Negative and Even";
    echo "<br>";
  }
  else{
        echo "$x is Negative and Odd";
    echo "<br>";
  }
}
?>
```

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
<?php
function name_palindrome($myname)
{
  if ($myname == strrev($myname))
     return 1;
  else
          return 0;
}
$name1 = 'anna';
if (name_palindrome($name1)){
   echo "$name1 is a Palindrome";
}
else {
        echo "$name1 is not a Palindrome";
}
echo "<br>";
$name2 = 'bogart';
if (name_palindrome($name2)){
   echo "$name2 is a Palindrome";
}
else {
        echo "$name2 is 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
<?php
function compute_factorial($a){
   $factorial = 1;
   for ($x = 1; $x <= $a; $x++){
     $factorial = $factorial * $x;
   }
   return $factorial;
}

// Driver Code
$test1 = 4;
$factor = compute_factorial($test1);
echo "Output = $factor";
?>
```

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
<?php
$a = 3;
$count = 1;
for ($x = $a; $x > 0; $x--)
{
  for ($y = $x; $y < $a + 1; $y++)
  {
    printf("%4s", $count);
    $count++;
  }
        echo "<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. array_column() array_combine()

	array_diff() array_keys() array_merge()
Calendar	The calendar extension contains functions that simplifies converting between different calendar formats. cal_days_in_month() cal_info() easter_date() jddayofweek() jdmonthname()
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. date.timezone date.default_latitude date.default_longitude date.sunrise_zenith date.sunset_zenith
Directory	The directory functions allow you to retrieve information about directories and their contents. chroot() dir() chdir() getcwd() closedir()
Error	The error functions allow us to define own error handling rules, and modify the way the errors can be logged. error_reporting display_errors log_errors track_errors html_errors
File System	The filesystem functions allow you to access and manipulate the filesystem. from allow_url_fopen sys_temp_dir allow_url_include user_agent
Filter	

	This PHP filters is used to validate and filter data coming from insecure sources, like user input. filter_input() filter_id() filter_list() filter_var() filter_var_array()
FTP	The FTP functions give client access to file servers through the File Transfer Protocol (FTP). ftp_alloc() ftp_chdir() ftp_close() ftp_delete() ftp_get()
Libxml	The libxml functions and constants are used together with SimpleXML, XSLT and DOM functions. libxml_clear_errors() libxml_get_errors() libxml_get_last_error() libxml_set_streams_context() libxml_use_internal_errors()
Mail	The mail() function allows you to send emails directly from a script. mail.log mail.add_x_header smtp_port sendmail_from sendmail_path
Math	The math functions can handle values within the range of integer and float types. base_convert() cos() log() max() min
Misc	The misc. functions are part of the PHP core. No installation is required to use these functions. highlight.string highlight.keyword highlight.comment highlight.default highlight.html

MySQLi	The MySQLi functions allows you to access MySQL database servers. autocommit() change_user() close() connect() debug()
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. closelog() dns_get_mx() dns_get_record() define_syslog_variables() dns_check_record()
SimpleXML	SimpleXML is an extension that allows us to easily manipulate and get XML data. __construct() addAttribute() attributes() children() count()
Stream	The Stream functions 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. A wrapper is additional code which tells the stream how to handle specific protocols/encodings. set_socket_blocking() stream_context_create() stream_context_get_default() stream_context_set_default() stream_context_set_options()
String	A string is a collection of characters. String is one of the data types supported by PHP.

	addcslashes() addslashes() chop() crypt() echo()
XML Parser	The XML functions lets you parse, but not validate, XML documents. xml_error_string() xml_parse() utf8_decode() utf8_encode() xml_parser_create_ns()
Zip	The Zip files functions allows you to read ZIP files. zip_open() zip_close() zip_read() zip_entry_open() zip_entry_read()
Timezones	The date_default_timezone_set() function is an inbuilt function in PHP which is used to set the default timezone used by all date/time functions in a script. This function returns False if the timezone is not valid, or True otherwise.

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.

   When creating a custom HTML template. Regular expressions can be used to identify the template tags and replace them with actual data.

   Example syntax in PHP:

   ```php
   <?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

   ```php
   <?php
   $sentence = "The quick brown fox";
   $y = "/Fox/i";
   if (preg_match($y, $sentence))
   {
           echo "Fox is found in the string";
   }
   else
   {
           echo "Fox 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
   <?php
   $sentence = "The quick brown fox";
   echo preg_replace('/\W\w+\s*(\W*)$/', '$1', $sentence)."\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
<?php
$mix = "/$123,34.00A#";
echo preg_replace("/[^0-9,.]/", "", $mix)."\n";
?>
```

       d. Write a PHP script to extract text (within parenthesis) from a string.
         Sample String: 'The quick brown [fox].'
         Expected output: Fox

```php
<?php
$sentence = 'The quick brown [fox].';
preg_match('#\[(.*?)\]#', $sentence, $matched);
print $matched[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
<?php
$letters = 'abcde$ddfd @abcd  )der]';
$x = preg_replace("/[^A-Za-z0-9 ]/", '', $letters);
echo 'Output : '.$x."\n";
?>
```

Activity 4: Error Handling

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

    a.) Parse error or Syntax Error: It is the type of error done by the programmer in the source code of the program.

EXAMPLE:

       `<?php include("csharpcorner.php"); ?>`

    b.) Fatal Error: It is the type of error where PHP compiler understand the PHP code but it recognizes an undeclared function.

EXAMPLE:

```
function shutDownFunction() {
    $error = error_get_last();
    // fatal error, E_ERROR === 1
```

```php
        if ($error['type'] === E_ERROR) {
            //do your stuff
        }
    }
    register_shutdown_function('shutDownFunction');
```

c.) Warning Errors : The main reason of warning errors are including a missing file. This means that the PHP function call the missing file.

d.) Notice Error: It is similar to warning error. It means that the program contains something wrong but it allows the execution of script.

EXAMPLE:

```php
<?php
if(file_exists("mytestfile.txt")) {
  $file = fopen("mytestfile.txt", "r");
} else {
  die("Error: The file does not exist.");
}
?>
```