**ACTIVITY ANSWER SHEET**

|  |  |
| --- | --- |
| Name | Adriane D. Jimenez |
| 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 | if (*condition*) { *//Code to be executed*; } |
| 2. if…else | if (*condition*) {     //C*ode to be executed if condition is* ***true****;* } else {   *//Code to be executed if condition is* ***false****;* } |
| 3. if…else if…else | if (*condition1*) {     //C*ode to be executed if this condition is* ***true****;* } elseif (*condition2*) {   *//Code to be executed if condition1 is* ***false*** *and condition2 is* ***true****;* } else {     //C*ode to be executed if both condition1 and condition2 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 (*initialization; condition; increment*) {   *//Code to be executed;* } |
| 6. do while loop | do { *execute the statements;* }  while (*condition is true*); |
| 7. while loop | while (*condition is true*) { *//Code to be executed*; } |
| 8. foreach loop | foreach ($*array\_expression*as$*value*) {   //C*ode to be executed;* } |
| 9. break statement | Jump statement;  break; |
| 10. continue statement | If (condition) {  //Code to be executed  continue;  } |
| 11. try…catch | try {  //code to be executed  }  catch (exception $a) {  //code to handle the exception  } |

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  $notnumber = '21';  $number = 21;  if (is\_int($number)) {  echo "$number is integer\n";  } else {  echo "$number is not an integer";  }  echo "<br>";  if (is\_int($notnumber)) {  echo "$notnumber is integer\n";  } else {  echo "$notnumber 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  $numberr = -21;  if ($numberr >= 0){  if ($numberr % -21 == 0){  echo "$numberr is positive and even";  echo "<br>";  }  else{  echo "$numberr is positive and odd";  echo "<br>";  }  }  else{  if ($numberr % -21 == 0){  echo "$numberr is negative and even";  echo "<br>";  }  else{  echo "$numberr 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  function palindrome($string)  {  if ($string == strrev($string))  return 1;  else  return 0;  }  $a = 'boob';  if (palindrome($a)){  echo "$a is a Palindrome";  }  else {  echo "$a is not a Palindrome";  }  echo "<br>";  $b = 'earwax';  if (palindrome($b)){  echo "$b is a Palindrome";  }  else {  echo "$b 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  function c\_factorial($numberr){  $f = 1;  for ($num = 1; $numm <= $numberr; $num++){  $f = $f \* $num;  }  return $f;  }    $a = 4;  $factor = c\_factorial($a);  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  $numberr = 3;  $start = 1;  for ($a = $numberr; $a > 0; $a--)  {  for ($b = $a; $b < $numberr + 1; $b++)  {  printf("%4s", $start);  $start++;  }  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 upercase  **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  **Array\_diff()** Compare arrays, ad return the differences  **Array\_map()** Sends each value of an array to a user-made function, which return new values  **Array\_merge()** Merges one or more arrays into one array  **Array\_multisort()**  sorts multiple or muti-dimensional arrays  **Array\_pad()** inserts a specified number of items, with a specified value, to an array  **Array\_pop()** deletes the last element of an array  **Array\_push()** inserts one or more elements to the end of an array  **Array\_reduce()** return an array as a string, using a user defined function  **array\_fill()** Fills an array with values  **array\_fill\_keys()** Fills an array with values, specifying keys  **array\_filter()** Filters the values of an array using a callback function  **array\_flip()** Flips/Exchanges all keys with their associated values in an array  **array\_intersect()** Compare arrays, and returns the matches (compare values only)  **Array\_reduce()** return an array as string, using a user-defined function  **Array\_search()** searches an array for a given value and returns the key  **Array\_shift()** removes the first element from an array, and return the value of the removed element  **Array\_slice()** reutrns selected parts of an array |
| Calendar | **cal\_days\_in\_month()** Returns the number of days in a month for a specified year and calendar  **easter\_days()** Returns the number of days after March 21, that the Easter Day is in a specified year  **frenchtojd()** Converts a French Republican date to a Julian Day Count  **gregoriantojd()** Converts a Gregorian date to a Julian Day Count  **jddayofweek()** Returns the day of the **weekcal\_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  **jdtojewish()** Converts a Julian Day Count to a Jewish date  **jdtojulian()** Converts a Julian Day Count to a Julian date  **jdtounix()** Converts Julian Day Count to Unix timestamp  **jewishtojd()** Converts a Jewish date to a Julian Day Count  **juliantojd()** Converts a Julian date to a Julian Day Count  **unixtojd()** Converts Unix timestamp to Julian Day Count |
| 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  **date\_default\_timezone\_get()** Returns the default timezone used by all date/time functions  **date\_default\_timezone\_set()** Sets the default timezone used by all date/time functions  **date\_diff()** Returns the difference between two dates  **date\_format()** Returns a date formatted according to a specified format  **date\_get\_last\_errors()** Returns the warnings/errors found in a date string  **date\_interval\_create\_from\_date\_string()** Sets up a DateInterval from the relative parts of the string  **date\_interval\_format()** Formats the interval  **date\_isodate\_set()** Sets the ISO date  **date\_modify()** Modifies the timestamp  **date\_offset\_get()** Returns the timezone offset  **date\_parse\_from\_format()** Returns an associative array with detailed info about a specified date, according to a specified format  **date\_parse()** Returns an associative array with detailed info about a specified date  **date\_sub()** Subtracts days, months, years, hours, minutes, and seconds from a date  **date\_sun\_info()** Returns an array containing info about sunset/sunrise and twilight begin/end, for a specified day and location  **date\_sunrise()** Returns the sunrise time for a specified day and location  **date\_sunset()** Returns the sunset time for a specified day and location  **date\_time\_set()** Sets the time  **date\_timestamp\_get()** Returns the Unix timestamp  **date\_timestamp\_set()** Sets the date and time based on a Unix timestamp  **date\_timezone\_get()** Returns the time zone of the given DateTime object  **date\_timezone\_set()** Sets the time zone for the DateTime object  **date()** Formats a local date and time  **getdate()** Returns date/time information of a timestamp or the current local date/time  **gettimeofday()** Returns the current time  **gmdate()** Formats a GMT/UTC date and time  **gmmktime()** Returns the Unix timestamp for a GMT date  **gmstrftime()** Formats a GMT/UTC date and time according to locale settings  **idate()** Formats a local time/date as integer  **localtime()** Returns the local time  **microtime()** Returns the current Unix timestamp with microseconds  **mktime()** Returns the Unix timestamp for a date  **strftime()** Formats a local time and/or date according to locale settings  **strptime()** Parses a time/date generated with strftime()  **strtotime()** Parses an English textual datetime into a Unix timestamp  **time()** Returns the current time as a Unix timestamp  **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  **timezone\_offset\_get()** Returns the timezone offset from GMT  **timezone\_open()** Creates new DateTimeZone object  **timezone\_transitions\_get()** Returns all transitions for the timezone  **timezone\_version\_get()** Returns the version of the timezonedb |
| 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  **opendir()** Opens a directory handle  **readdir()** Returns an entry from a directory handle  **rewinddir()** Resets a directory handle  **scandir()** Returns an array of files and directories of a specified directory |
| Error | **debug\_backtrace()** Generates a backtrace  **debug\_print\_backtrace()** Prints a backtrace  **error\_clear\_last()** Clears the last error  **error\_get\_last()** Returns the last error that occurred  **error\_log()** Sends an error message to a log, to a file, or to a mail account  **error\_reporting()** Specifies which errors are reported  r**estore\_error\_handler()** Restores the previous error handler  **restore\_exception\_handler()** Restores the previous exception handler  **set\_error\_handler()** Sets a user-defined error handler function  **set\_exception\_handler()** Sets a user-defined exception handler function  **trigger\_error()** Creates a user-level error message  **user\_error()** Alias of trigger\_error() |
| File System | **allow\_url\_fopen** "1" Allows fopen()-type functions to work with URLs PHP\_INI\_SYSTEM  **allow\_url\_include** "0" (available since PHP 5.2) PHP\_INI\_SYSTEM  **user\_agent** NULL Defines the user agent for PHP to send (available since PHP 4.3) PHP\_INI\_ALL  **default\_socket\_timeout** "60" Sets the default timeout, in seconds, for socket based streams (available since PHP 4.3) PHP\_INI\_ALL  **sys\_temp\_dir** "" (available since PHP 5.5) PHP\_INI\_SYSTEM  **basename()** Returns the filename component of a path  **chgrp()** Changes the file group  **chmod()** Changes the file mode  **chown()** Changes the file owner  **clearstatcache()** Clears the file status cache  **copy()** Copies a file  **delete()** Delete a File  **dirname()** Returns the directory name component of a path |
| 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  **filter\_var()** Filters a variable with a specified filter  **filter\_var\_array()** Gets multiple variables and filter them |
| FTP | **ftp\_login()** Logs in to the FTP connection  **ftp\_mdtm()** Returns the last modified time of a specified file  **ftp\_mkdir()** Creates a new directory on the FTP server  **ftp\_mlsd()** Returns the list of files in the specified directory  **ftp\_nb\_continue()** Continues retrieving/sending a file (non-blocking) |
| Libxml | **libxml\_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 | **decbin()** Converts a decimal number to a binary number  **dechex()** Converts a decimal number to a hexadecimal number  **decoct()** Converts a decimal number to an octal number  **deg2rad()** Converts a degree value to a radian value  **exp()** Calculates the exponent of e  **pi()** Returns the value of PI  **pow()** Returns x raised to the power of y  **rad2deg()** Converts a radian value to a degree value  **rand()** Generates a random integer  **round()** Rounds a floating-point number  **sin()** Returns the sine of a number  **sinh()** Returns the hyperbolic sine of a number  **sqrt()** Returns the square root of a number  **srand()** Seeds the random number generator  t**an()** Returns the tangent of a number  **tanh()** Returns the hyperbolic tangent of a number |
| Misc | **connection\_aborted()** Checks whether the client has disconnected  **connection\_status()** Returns the current connection status  **connection\_timeout()** Deprecated from PHP 4.0.5. Checks whether the script has timed out  **defined()** Checks whether a constant exists  **die()**  **Alias of exit()**  **eval()** Evaluates a string as PHP code  **exit()** Prints a message and exits the current script  **get\_browser()** Returns the capabilities of the user's browser |
| MySQLi | **errno()** Returns the last error code for the most recent function call  **error()** Returns the last error description for the most recent function call  **error\_list()** Returns a list of errors for the most recent function call  **fetch\_all()** Fetches all result rows as an associative array, a numeric array, or both  **fetch\_array()** Fetches a result row as an associative, a numeric array, or both |
| Network | **getprotobyname()** Returns the protocol number for a given protocol name  **getprotobynumber()** Returns the protocol name for a given protocol number  **getservbyname()** Returns the port number for a given Internet service and protocol  **getservbyport()** Returns the Internet service for a given port and protocol  **header\_register\_callback()** Calls a header function |
| SimpleXML | **getDocNamespaces()** Returns the namespaces declared in document  **getName()** Returns the name of an element  **getNamespaces()** Returns the namespaces used in document  **registerXPathNamespace()** Creates a namespace context for the next XPath query  **saveXML()** Alias of asXML() |
| Stream | **stream\_context\_get\_options()**  **stream\_context\_get\_params()**  **stream\_context\_set\_default()**  **stream\_context\_set\_options()**  **stream\_context\_set\_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  c**hop()** Removes whitespace or other characters from the right end of a string  **chr()** Returns a character from a specified ASCII value  **chunk\_split()** Splits a string into a series of smaller parts  **convert\_cyr\_string()** Converts a string from one Cyrillic character-set to another  **convert\_uudecode()** Decodes a uuencoded string  **convert\_uuencode()** Encodes a string using the uuencode algorithm  **count\_chars()** Returns information about characters used in a string  **crc32()** Calculates a 32-bit CRC for a string |
| XML Parser | **xml\_get\_current\_column\_number()**Returns the current column number from the XML parser  **xml\_get\_current\_line\_number()** Returns the current line number from the XML parser  **xml\_get\_error\_code()** Returns an error code from the XML parser  **xml\_parse()** Parses an XML document  **xml\_parse\_into\_struct()** Parses XML data into an array |
| Zip | **zip\_entry\_name()** Returns the name of a ZIP directory entry  **zip\_entry\_open()** Opens a directory entry in a ZIP file for reading  **zip\_entry\_read()** Reads from an open directory entry in the ZIP file  **zip\_open()** Opens a ZIP file archive  **zip\_read()** Reads the next file in a open ZIP file archive |
| Timezones | **PHP Date/Time Functions**  **PHP gmdate() Function**  **PHP strtotime() Function**  **PHP Date and Time**  **PHP Tryit Editor v1.1** |

**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.

<?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  $str = "The quick brown fox";  $test = "/Fox/i";  if (preg\_match($test, $str))  {  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  $str = "The quick brown fox";  echo preg\_replace('/\W\w+\s\*(\W\*)$/', '$1', $str)."\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  $str = "/$123,34.00A#";  echo preg\_replace("/[^0-9,.]/", "", $str)."\n";  ?> |

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

Sample String: ‘The quick brown [fox].’

Expected output: Fox

|  |
| --- |
| <?php  $str = 'The quick brown [fox].';  preg\_match('#\[(.\*?)\]#', $str, $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  $a = 'abcde$ddfd @abcd )der]';  $b = preg\_replace("/[^A-Za-z0-9 ]/", '', $a);  echo 'Output : '.$b."\n";  ?> |

**Activity 4: Error Handling**

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

1. **Syntax Error**

<?php

$x = "qwe";

y = "qweqwe";

echo $x;

echo $y;

?> //WRONG//

<?php

$x = "qwe";

$y = "qweqwe";

echo $x;

echo $y;

?> //CORRECT! Observe always//

1. **Fatal Error**

<?php

function add($x, $y)

{

$sum = $x + $y;

echo "sum = " . $sum;

}

$x = 0;

$y = 20;

add($x, $y); //don’t call some functions that is not ?> availble. Always check!! grr!

1. **Warning Errors**

<?php

$x = "qwe";

include ("rofl.php"); //check the spelling maybe your calling the wrong php file. Which

echo $x . "qweqwe"; the computer could not find

?>

1. **Notice Error**

<?php

$x = "GeeksforGeeks";

echo $x;

echo $geeks; //don’t call things u didn’t assign grr!

?>

<?php

$x = "GeeksforGeeks";

echo $x; //perfection! ;)

?>