**ACTIVITY ANSWER SHEET**

|  |  |
| --- | --- |
| Name | Efren Ray C. Pineda |
| Section: | BSIT 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) {  executed if condition is true; } |
| 2. if…else | if (condition){  executed if condition is true; } else {  executed if condition is false; } |
| 3. if…else if…else | if (condition) {  executed if this condition is true; } elseif (condition) {  executed if first condition is false and this condition is true; } else { executed if all conditions are false; } |
| 4. switch…case | switch (n) { case nlabel1:  executed if n=label1;  break;  case label2:  executed if n=label2; break;  case label3:  executed if n=label3; break;  ...  default:  executed if n is different from all labels; } |
| 5. for loop | for (init counter; test counter; increment counter) { executed for each iteration; } |
| 6. do while loop | do {  executed; } while (condition is true); |
| 7. while loop | while (condition is true) {  code to be executed; } |
| 8. foreach loop | foreach ($array as $value) { code to be executed; } |
| 9. break statement | statement;  break; |
| 10. continue statement | Loop (While, do-while, for,) { conditions { continue;//continue statement } code executed; } |
| 11. try…catch | function checkNum($number) {  if($number>1) {  throw new Exception("Value must be 1 or below");  }  return true;  }  try {  checkNum(2);  echo 'If you see this, the number is 1 or below';  }  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

|  |
| --- |
| if (is\_numeric($input)) {  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

|  |
| --- |
| if ($input >= 0 && $input % 2 == 0) {  echo “Positive & Even”;  } else if ($input >= 0 && $input % 2 !=0){  echo “Positive & Odd”;  } else if ($input < 0 && $input % 2 == 0) {  echo “Negative & Even”;  } else {  echo “Negative & Odd”;  } |

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

|  |
| --- |
| function Palindrome($string){      if (strrev($string) == $string){          return 1;      }      else{          return 0;      }  }    // Driver Code  $original = "DAD";  if(Palindrome($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

|  |
| --- |
| $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

|  |
| --- |
| $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()  array\_change\_key\_case()  array\_chunk()  array\_column()  array\_combine()  array\_count\_values()  array\_diff()  array\_diff\_assoc()  array\_diff\_key()  array\_diff\_uassoc()  array\_diff\_ukey()  array\_fill()  array\_fill\_keys()  array\_filter()  array\_flip()  array\_intersect()  array\_intersect\_assoc()  array\_intersect\_key()  array\_intersect\_uassoc()  array\_intersect\_ukey()  array\_key\_exists()  array\_keys()  array\_map()  array\_merge()  array\_merge\_recursive()  array\_multisort()  array\_pad()  array\_pop()  array\_product()  array\_push()  array\_rand()  array\_reduce()  array\_replace()  array\_replace\_recursive()  array\_reverse()  array\_search()  array\_shift()  array\_slice()  array\_splice()  array\_sum()  array\_udiff()  array\_udiff\_assoc()  array\_udiff\_uassoc()  array\_uintersect()  array\_uintersect\_assoc()  array\_uintersect\_uassoc()  array\_unique()  array\_unshift()  array\_values()  array\_walk()  array\_walk\_recursive()  arsort()  asort()  compact()  count()  current()  each()  end()  extract()  in\_array()  key()  krsort()  ksort()  list()  natcasesort()  natsort()  next()  pos()  prev()  range()  reset()  rsort()  shuffle()  sizeof()  sort()  uasort()  uksort()  usort() |
| Calendar | cal\_days\_in\_month()  cal\_from\_jd()  cal\_info()  cal\_to\_jd()  easter\_date()  easter\_days()  frenchtojd()  gregoriantojd()  jddayofweek()  jdmonthname()  jdtofrench()  jdtogregorian()  jdtojewish()  jdtojulian()  jdtounix()  jewishtojd()  juliantojd()  unixtojd() |
| Date | checkdate()  date\_add()  date\_create\_from\_format()  date\_create()  date\_date\_set()  date\_default\_timezone\_get()  date\_default\_timezone\_set()  date\_diff()  date\_format()  date\_get\_last\_errors()  date\_interval\_create\_from\_date\_string()  date\_interval\_format()  date\_isodate\_set()  date\_modify()  date\_offset\_get()  date\_parse\_from\_format()  date\_parse()  date\_sub()  date\_sun\_info()  date\_sunrise()  date\_sunset()  date\_time\_set()  date\_timestamp\_get()  date\_timestamp\_set()  date\_timezone\_get()  date\_timezone\_set()  date()  getdate()  gettimeofday()  gmdate()  gmmktime()  gmstrftime()  idate()  localtime()  microtime()  mktime()  strftime()  strptime()  strtotime()  time()  timezone\_abbreviations\_list()  timezone\_identifiers\_list()  timezone\_location\_get()  timezone\_name\_from\_ abbr()  timezone\_name\_get()  timezone\_offset\_get()  timezone\_open()  timezone\_transitions\_get()  timezone\_version\_get() |
| Directory | chdir()  chroot()  closedir()  dir()  getcwd()  opendir()  readdir()  rewinddir()  scandir() |
| Error | debug\_backtrace()  debug\_print\_backtrace()  error\_clear\_last()  error\_get\_last()  error\_log()  error\_reporting()  restore\_error\_handler()  restore\_exception\_handler()  set\_error\_handler()  set\_exception\_handler()  trigger\_error()  user\_error() |
| File System | basename()  chgrp()  chmod()  chown()  clearstatcache()  copy()  delete()  dirname()  disk\_free\_space()  disk\_total\_space()  diskfreespace()  fclose()  feof()  fflush()  fgetc()  fgetcsv()  fgets()  fgetss()  file()  file\_exists()  file\_get\_contents()  file\_put\_contents()  fileatime()  filectime()  filegroup()  fileinode()  filemtime()  fileowner()  fileperms()  filesize()  filetype()  flock()  fnmatch()  fopen()  fpassthru()  fputcsv()  fputs()  fread()  fscanf()  fseek()  fstat()  ftell()  ftruncate()  fwrite()  glob()  is\_dir()  is\_executable()  is\_file()  is\_link()  is\_readable()  is\_uploaded\_file()  is\_writable()  is\_writeable()  lchgrp()  lchown()  link()  linkinfo()  lstat()  mkdir()  move\_uploaded\_file()  parse\_ini\_file()  parse\_ini\_string()  pathinfo()  pclose()  popen()  readfile()  readlink()  realpath()  realpath\_cache\_get()  realpath\_cache\_size()  rename()  rewind()  rmdir()  set\_file\_buffer()  stat()  symlink()  tempnam()  tmpfile()  touch()  umask()  unlink() |
| Filter | filter\_has\_var()  filter\_id()  filter\_input()  filter\_input\_array()  filter\_list()  filter\_var()  filter\_var\_array() |
| FTP | ftp\_alloc()  ftp\_cdup()  ftp\_chdir()  ftp\_chmod()  ftp\_close()  ftp\_connect()  ftp\_delete()  ftp\_exec()  ftp\_fget()  ftp\_fput()  ftp\_get()  ftp\_get\_option()  ftp\_login()  ftp\_mdtm()  ftp\_mkdir()  ftp\_mlsd()  ftp\_nb\_continue()  ftp\_nb\_fget()  ftp\_nb\_fput()  ftp\_nb\_get()  ftp\_nb\_put()  ftp\_nlist()  ftp\_pasv()  ftp\_put()  ftp\_pwd()  ftp\_quit()  ftp\_raw()  ftp\_rawlist()  ftp\_rename()  ftp\_rmdir()  ftp\_set\_option()  ftp\_site()  ftp\_size()  ftp\_ssl\_connect()  ftp\_systype() |
| Libxml | libxml\_clear\_errors()  libxml\_disable\_entity\_loader()  libxml\_get\_errors()  libxml\_get\_last\_error()  libxml\_set\_external\_entity\_loader()  libxml\_set\_streams\_context()  libxml\_use\_internal\_errors() |
| Mail | ezmlm\_hash()  mail() |
| Math | abs()  acos()  acosh()  asin()  asinh()  atan()  atan2()  atanh()  base\_convert()  bindec()  ceil()  cos()  cosh()  decbin()  dechex()  decoct()  deg2rad()  exp()  expm1()  floor()  fmod()  getrandmax()  hexdec()  hypot()  intdiv()  is\_finite()  is\_infinite()  is\_nan()'  lcg\_value()  log()  log10()  log1p()  max()  min()  mt\_getrandmax()  mt\_rand()  mt\_srand()  octdec()  pi()  pow()  rad2deg()  rand()  round()  sin()  sinh()  sqrt()  srand()  tan()  tanh() |
| Misc | connection\_aborted()  connection\_status()  connection\_timeout()  constant()  define()  defined()  die()  eval()  exit()  get\_browser()  \_\_halt\_compiler()  highlight\_file()  highlight\_string()  hrtime()  ignore\_user\_abort()  pack()  php\_strip\_whitespace()  show\_source()  sleep()  sys\_getloadavg()  time\_nanosleep()  time\_sleep\_until()  uniqid()  unpack()  unpack()  usleep() |
| MySQLi | affected\_rows()  autocommit()  begin\_transaction()  change\_user()  character\_set\_name()  close()  commit()  connect()  connect\_errno()  connect\_error()  data\_seek()  debug()  dump\_debug\_info()  errno()  error()  error\_list()  fetch\_all()  fetch\_array()  fetch\_assoc()  fetch\_field()  fetch\_field\_direct()  fetch\_fields()  fetch\_lengths()  fetch\_object()  fetch\_row()  field\_count()  field\_seek()  get\_charset()  get\_client\_info()  get\_client\_stats()  get\_client\_version()  get\_connection\_stats()  get\_host\_info()  get\_proto\_info()  get\_server\_info()  get\_server\_version()  info()  init()  insert\_id()  kill()  more\_results()  multi\_query()  next\_result()  options()  ping()  poll()  prepare()  query()  real\_connect()  real\_escape\_string()  real\_query()  reap\_async\_query()  refresh()  rollback()  select\_db()  set\_charset()  set\_local\_infile\_default()  set\_local\_infile\_handler()  sqlstate()  ssl\_set()  stat()  stmt\_init()  store\_result()  thread\_id()  thread\_safe()  use\_result()  warning\_count() |
| Network | checkdnsrr()  closelog()  define\_syslog\_variables()  dns\_check\_record()  dns\_get\_mx()  dns\_get\_record()  fsockopen()  gethostbyaddr()  gethostbyname()  gethostbynamel()  gethostname()  getmxrr()  getprotobyname()  getprotobynumber()  getservbyname()  getservbyport()  header\_register\_callback()  header\_remove()  header()  headers\_list()  headers\_sent()  http\_response\_code()  inet\_ntop()  inet\_pton()  ip2long()  long2ip()  openlog()  pfsockopen()  setcookie()  setrawcookie()  socket\_get\_status()  socket\_set\_blocking()  socket\_set\_timeout()  syslog() |
| SimpleXML | \_\_construct()  \_\_toString()  addAttribute()  addChild()  asXML()  attributes()  children()  count()  getDocNamespaces()  getName()  getNamespaces()  registerXPathNamespace()  saveXML()  simplexml\_import\_dom()  simplexml\_load\_file()  simplexml\_load\_string()  xpath() |
| Stream | set\_socket\_blocking()  stream\_bucket\_prepend()  stream\_context\_create()  stream\_context\_get\_default()  stream\_context\_get\_options()  stream\_context\_get\_params()  stream\_context\_set\_default()  stream\_context\_set\_options()  stream\_context\_set\_params()  stream\_copy\_to\_stream()  stream\_filter\_append()  stream\_filter\_prepend()  stream\_filter\_register()  stream\_filter\_remove()  stream\_get\_contents()  stream\_get\_filters()  stream\_get\_line()  stream\_get\_meta\_data()  stream\_get\_transports()  stream\_get\_wrappers()  stream\_is\_local()  stream\_isatty()  stream\_notification\_callback()  stream\_register\_wrapper()  stream\_resolve\_include\_path()  stream\_select()  stream\_set\_blocking()  stream\_set\_chunk\_size()  stream\_set\_read\_buffer()  stream\_set\_timeout()  stream\_set\_write\_buffer()  stream\_socket\_accept()  stream\_socket\_client()  stream\_socket\_enable\_crypto()  stream\_socket\_get\_name()  stream\_socket\_pair()  stream\_socket\_recvfrom()  stream\_socket\_sendto()  stream\_socket\_server()  stream\_socket\_shutdown()  stream\_supports\_lock()  stream\_wrapper\_register()  stream\_wrapper\_restore()  stream\_wrapper\_unregister() |
| String | addcslashes()  addslashes()  bin2hex()  chop()  chr()  chunk\_split()  convert\_cyr\_string()  convert\_uudecode()  convert\_uuencode()  count\_chars()  crc32()  crypt()  echo()  explode()  fprintf()  get\_html\_translation\_table()  hebrev()  hebrevc()  hex2bin()  html\_entity\_decode()  htmlentities()  htmlspecialchars\_decode()  htmlspecialchars()  implode()  join()  lcfirst()  levenshtein()  localeconv()  ltrim()  md5()  md5\_file()  metaphone()  money\_format()  nl\_langinfo()  nl2br()  number\_format()  ord()  parse\_str()  print()  printf()  quoted\_printable\_decode()  quoted\_printable\_encode()  quotemeta()  rtrim()  setlocale()  sha1()  sha1\_file()  similar\_text()  soundex()  sprintf()  sscanf()  str\_getcsv()  str\_ireplace()  str\_pad()  str\_repeat()  str\_replace()  str\_rot13()  str\_shuffle()  str\_split()  str\_word\_count()  strcasecmp()  strchr()  strcmp()  strcoll()  strcspn()  strip\_tags()  stripcslashes()  stripslashes()  stripos()  stristr()  strlen()  strnatcasecmp()  strnatcmp()  strncasecmp()  strncmp()  strpbrk()  strpos()  strrchr()  strrev()  strripos()  strrpos()  strspn()  strstr()  strtok()  strtolower()  strtoupper()  strtr()  substr()  substr\_compare()  substr\_count()  substr\_replace()  trim()  ucfirst()  ucwords()  vfprintf()  vprintf()  vsprintf()  wordwrap() |
| XML Parser | utf8\_decode()  utf8\_encode()  xml\_error\_string()  xml\_get\_current\_byte\_index()  xml\_get\_current\_column\_number()  xml\_get\_current\_line\_number()  xml\_get\_error\_code()  xml\_parse()  xml\_parse\_into\_struct()  xml\_parser\_create\_ns()  xml\_parser\_create()  xml\_parser\_free()  xml\_parser\_get\_option()  xml\_parser\_set\_option()  xml\_set\_character\_data\_handler()  xml\_set\_default\_handler()  xml\_set\_element\_handler()  xml\_set\_end\_namespace\_decl\_handler()  xml\_set\_external\_entity\_ref\_handler()  xml\_set\_notation\_decl\_handler()  xml\_set\_object()  xml\_set\_processing\_instruction\_handler()  xml\_set\_start\_namespace\_decl\_handler()  xml\_set\_unparsed\_entity\_decl\_handler() |
| Zip | zip\_close()  zip\_entry\_close()  zip\_entry\_compressedsize()  zip\_entry\_compressionmethod()  zip\_entry\_filesize()  zip\_entry\_name()  zip\_entry\_open()  zip\_entry\_read()  zip\_open()  zip\_read() |
| Timezones | Africa  America  Antarctica  Arctic  Asia  Atlantic  Australia  Europe  Indian  Pacific |

**Activity 3: Regular Expression**

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

$my\_url = "www.efren.com";

if (preg\_match("/efren/", $my\_url))

{

echo "the url $my\_url contains efren";

}

else

{

echo "the url $my\_url does not contain efren";

}

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

|  |
| --- |
| $str1 = 'The quick brown fox.';  if (strpos($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'

|  |
| --- |
| $str1 = 'The quick brown fox';  echo preg\_replace('/\W\w+\s\*(\W\*)$/', '$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

|  |
| --- |
| $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

|  |
| --- |
| $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

|  |
| --- |
| $string = 'abcde$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.

**Parse error or Syntax Error**

-----

Example of a non-indented code:

<?

if ($condition){

echo "true";

?>

This is often due to a poorly organized presentation of your code.

Especially remember to indent your code well, to visually distinguish the different blocks.

**Notice Error**

----

Example of notice error:

<?php

$x = "GeeksforGeeks";

echo $x;

echo $geeks;

?>

This program use undeclared variable $geeks so it gives error message.

**Warning Error**

--------

Example of warning error:

<?php

$x = "GeeksforGeeks";

include ("gfg.php");

echo $x . "Computer science portal";

?>

This program call an undefined file gfg.php which are not available. So it produces error.

**Fatal Error**

----------

Example of fatal error :

<?php

function add($x, $y)

{

$sum = $x + $y;

echo "sum = " . $sum;

}

$x = 0;

$y = 20;

add($x, $y);

diff($x, $y);

?>

In line 12, function is called but the definition of function is not available. So it gives error.