

1000-php-questions-answers

- a) Both i) and iii)
- b) Both ii) and iv)
- c) Only ii)
- d) Both i) and ii)

Answer: d

Explanation: PHP previously stood for Personal Home Page now stands for Hypertext Preprocessor.

2. PHP files have a default file extension of _____

- a) .html
- b) .xml
- c) .php
- d) .ph

Answer: c

Explanation: To run a php file on the server, it should be saved as AnyName.php

3. What should be the correct syntax to write a PHP code?

- a) < php >
- b) < ? php ?>
- c) <? ?>
- d) <?php ?>

Answer: c

Explanation: Every section of PHP code starts and ends by turning on and off PHP tags to let the server know that it needs to execute the PHP in between them.

- a) Only iv)
- b) i), ii), iii) and iv)
- c) i), ii) and iii)
- d) Only iii)

Answer: b

Explanation: Any of the above editors can be used to type php code and run it.

- a) i), ii), iii) and iv)
- b) Only ii)
- c) ii) and iii)
- d) ii), iii) and iv)

Answer: d

Explanation: To run PHP code you need to have PHP and a web server, both IIS and Apache are web servers. You can choose either one according to your platform.

6. Which version of PHP introduced Try/catch Exception?

- a) PHP 4
- b) PHP 5
- c) PHP 6
- d) PHP 5 and later

Answer: d

Explanation: PHP 5 version and later versions added support for Exception Handling.

- a) Only ii)
- b) i), iii) and iv)

- c) ii), iii) and iv)
- d) Both ii) and iv)

Answer: c

Explanation: / */ can also be use to comment just a single line although it is used for paragraphs. // and # are used only for single line comment.*

8. Which of the following PHP statement/statements will store 111 in variable num?

```
i) int $num = 111;
ii) int num = 111;
iii) $num = 111;
iv) 111 = $num;
```

- a) Both i) and ii)
- b) i), ii), iii) and iv)
- c) Only iii)
- d) Only i)

Answer: c

Explanation: You need not specify the datatype in php.

9. What will be the output of the following PHP code?

```
1.      <?php
2.      $num  = 1;
3.      $num1 = 2;
4.      print $num . "+" . $num1;
5.      ?>
```

- a) 3
- b) 1+2
- c) 1+.2
- d) Error

Answer: b

Explanation: .(dot) is used to combine two parts of the statement. Example (\$num . "Hello World") will output 1Hello World.

10. What will be the output of the following PHP code?

```
1.      <?php
2.      $num  = "1";
3.      $num1 = "2";
4.      print $num+$num1;
5.      ?>
```

- a) 3
- b) 1+2
- c) Error
- d) 12

Answer: a

Explanation: The numbers inside the double quotes are considered as integers and not string, therefore the value 3 is

printed and not 1+2.

1. Which is the right way of declaring a variable in PHP?

```
i) $3hello  
ii) $_hello  
iii) $this  
iv) $This
```

- a) Only ii)
- b) Only iii)
- c) ii), iii) and iv)
- d) ii) and iv)

Answer: d

Explanation: A variable in PHP can not start with a number, also \$this is mainly used to refer properties of a class so we can't use \$this as a user define variable name.

2. What will be the output of the following PHP code?

```
1.      <?php  
2.      $foo = 'Bob';  
3.      $bar = &$foo;  
4.      $bar = "My name is $bar";  
5.      echo $bar;  
6.      echo $foo;  
7.      ?>
```

- a) Error
- b) My name is BobBob
- c) My name is BobMy name is Bob
- d) My name is Bob Bob

Answer: c

Explanation: Firstly, the line \$bar = &\$foo; will reference \$foo via \$bar. So \$bar is assigned value Bob. Therefore \$bar = "My name is \$bar"; will print My name is Bob(\$bar=Bob as said before).

3. Which of the following PHP statements will output Hello World on the screen?

```
i) echo ("Hello World");  
ii) print ("Hello World");  
iii) printf ("Hello World");  
iv) sprintf ("Hello World");
```

- a) i) and ii)
- b) i), ii) and iii)
- c) i), ii), iii) and iv)
- d) i), ii) and iv)

Answer: b

Explanation: echo(), print() and printf() all three can be used to output a statement onto the screen. The sprintf() statement is functionally identical to printf() except that the output is assigned to a string rather than rendered to the browser.

4. What will be the output of the following PHP code?

```
1.      <?php
2.      $color = "maroon";
3.      $var = $color[2];
4.      echo "$var";
5.      ?>
```

- a) a
- b) Error
- c) \$var
- d) r

Answer: d

Explanation: PHP treats strings in the same fashion as arrays, allowing for specific characters to be accessed via array offset notation. In an array, index always starts from 0. So in the line \$var = \$color[2]; if we count from start 'r' comes at index 2. So the output will be r.

5. What will be the output of the following PHP code?

```
1.      <?php
2.      $score = 1234;
3.      $scoreboard = (array) $score;
4.      echo $scoreboard[0];
5.      ?>
```

- a) 1
- b) Error
- c) 1234
- d) 2

Answer: c

Explanation: The (array) is a cast operator which is used for converting values from other data types to array.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      $total = "25 students";
3.      $more = 10;
4.      $total = $total + $more;
5.      echo "$total";
6.      ?>
```

- a) Error
- b) 35 students
- c) 35
- d) 25 students

Answer: c

Explanation: The integer value at the beginning of the original \$total string is used in the calculation. However if it begins with anything but a numerical value, the value will be 0.

7. Which of the below statements is equivalent to \$add += \$add?

- a) \$add = \$add
- b) \$add = \$add + \$add
- c) \$add = \$add + 1
- d) \$add = \$add + \$add + 1

Answer: b

Explanation: $a += b$ is an addition assignment whose outcome is $a = a + b$. Same can be done with subtraction, multiplication, division etc.

8. Which statement will output \$x on the screen?

- a) echo “\ \$x”;
- b) echo “\$ \$x”;
- c) echo “/ \$x”;
- d) echo “\$x;”;

Answer: a

Explanation: A backslash is used so that the dollar sign is treated as a normal string character rather than prompt PHP to treat \$x as a variable. The backslash used in this manner is known as escape character.

9. What will be the output of the following PHP code?

```
1.      <?php
2.      function track() {
3.          static $count = 0;
4.          $count++;
5.          echo $count;
6.      }
7.      track();
8.      track();
9.      track();
10.     ?>
```

- a) 123
- b) 111
- c) 000
- d) 011

Answer: a

Explanation: Because \$count is static, it retains its previous value each time the function is executed.

10. What will be the output of the following PHP code?

```
1.      <?php
2.      $a = "clue";
3.      $a .= "get";
4.      echo "$a";
5.      ?>
```

- a) get
- b) true
- c) false
- d) clueget

Answer: d

Explanation: '.' is a concatenation operator. \$a. = "get" is same as \$a=\$a."get" where \$a is having value of "clue" in the previous statement. So the output will be clueget.

1. What will be the output of the following PHP code?

```
1.      <?php
2.      $a = 5;
3.      $b = 5;
4.      echo ($a === $b);
5.      ?>
```

- a) 5 === 5
- b) Error
- c) 1
- d) False

Answer: c

Explanation: === operator returns 1 if \$a and \$b are equivalent and \$a and \$b have the same type.

2. Which of the below symbols is a newline character?

- a) \r
- b) \n
- c) /n
- d) /r

Answer: b

Explanation: PHP treats \n as a newline character.

3. What will be the output of the following PHP code?

```
1.      <?php
2.      $num = 10;
3.      echo 'What is her age? \n She is $num years old';
4.      ?>
```

- a) Only i)
- b) i), ii) and iv)
- c) ii), iii) and iv)
- d) i), ii), iii) and iv)

Answer: d

Explanation: All are conditional statements supported by PHP as all are used to evaluate different conditions during a program and take decisions based on whether these conditions evaluate to true or false.

5. What will be the output of the following PHP code?

```
1.      <?php
```

```

2.      $team = "arsenal";
3.
4.      switch ($team) {
5.
6.          case "manu":
7.              echo "I love man u";
8.
9.          case "arsenal":
10.             echo "I love arsenal";
11.
12.         case "manc":
13.             echo "I love manc"; }
14.
15.     ?>

```

- a) I love arsenal
- b) Error
- c) I love arsenalI love manc
- d) I love arsenalI love mancI love manu

Answer: c

Explanation: If a break statement isn't present, all subsequent case blocks will execute until a break statement is located.

- a) i) and ii)
- b) i), ii) and iii)
- c) i), ii), iii) and iv)
- d) Only iv)

Answer: c

Explanation: All are supported looping statements in PHP as they can repeat the same block of code a given number of times, or until a certain condition is met.

7. What will be the output of the following PHP code?

```

1.      <?php
2.
3.      $user = array("Ashley", "Bale", "Shrek", "Blank");
4.
5.      for ($x=0; $x < count($user); $x++) {
6.
7.          if ($user[$x] == "Shrek") continue;
8.
9.          printf ($user[$x]);
10.
11.      }
12.
13.     ?>

```

- a) AshleyBale
- b) AshleyBaleBlank
- c) ShrekBlank
- d) Shrek

Answer: b

Explanation: The continue statement causes execution of the current loop iteration to end and commence at the beginning of the next iteration.

8. If \$a = 12 what will be returned when (\$a == 12) ? 5 : 1 is executed?

- a) 12

- b) 1
- c) Error
- d) 5

Answer: d

Explanation: ?: is known as ternary operator. If condition is true then the part just after the ? is executed else the part after : .

9. What will be the value of \$a and \$b after the function call in the following PHP code?

```
1.      <?php
2.      function doSomething( &$arg ) {
3.          $return = $arg;
4.          $arg += 1;
5.          return $return;
6.      }
7.      $a = 3;
8.      $b = doSomething( $a );
9.      ?>
```

- a) a is 3 and b is 4
- b) a is 4 and b is 3
- c) Both are 3
- d) Both are 4

Answer: b

Explanation: \$a is 4 and \$b is 3. The former because \$arg is passed by reference, the latter because the return value of the function is a copy of the initial value of the argument.

10. Who is the father of PHP?

- a) Rasmus Lerdorf
- b) Willam Makepiece
- c) DreK Kolkevi
- d) List Barely

Answer: a

Explanation: PHP was originally created by Rasmus Lerdorf in 1994.

1. How to define a function in PHP?

- a) function {function body}
- b) data type functionName(parameters) {function body}
- c) functionName(parameters) {function body}
- d) function functionName(parameters) {function body}

Answer: d

Explanation: PHP allows us to create our own user-defined functions. Any name ending with an open and closed parenthesis is a function. The keyword function is always used to begin a function.

2. Type Hinting was introduced in which version of PHP?

- a) PHP 4
- b) PHP 5
- c) PHP 5.3

d) PHP 6

Answer: b

Explanation: PHP 5 introduced the feature of type hinting. With the help of type hinting, we can specify the expected data type of an argument in a function declaration. First valid types can be the class names for arguments that receive objects and the other are array for those that receive arrays.

3. Which type of function call is used in line 8 in the following PHP code?

```
1.      <?php
2.      function calc($price, $tax)
3.      {
4.          $total = $price + $tax;
5.      }
6.      $pricetag = 15;
7.      $taxtag = 3;
8.      calc($pricetag, $taxtag);
9.      ?>
```

- a) Call By Value
- b) Call By Reference
- c) Default Argument Value
- d) Type Hinting

Answer: a

Explanation: If we call a function by value, we actually pass the values of the arguments which are stored or copied into the formal parameters of the function. Hence, the original values are unchanged only the parameters inside the function changes.

4. What will be the output of the following PHP code?

```
1.      <?php
2.      function calc($price, $tax="")
3.      {
4.          $total = $price + ($price * $tax);
5.          echo "$total";
6.      }
7.      calc(42);
8.      ?>
```

- a) Error
- b) 0
- c) 42
- d) 84

Answer: c

Explanation: You can designate certain arguments as optional by placing them at the end of the list and assigning them a default value of nothing.

5. Which of the following are valid function names?

- i) function()
- ii) €()
- iii) .function()
- iv) \$function()

- a) Only i)
- b) Only ii)
- c) i) and ii)
- d) iii) and iv)

Answer: b

Explanation: A valid function name can start with a letter or underscore, followed by any number of letters, numbers, or underscores. According to the specified regular expression $([a-zA-Z_\\x7f-\\xff][a-zA-Z0-9_\\x7f-\\xff]^*)$, a function name like this one is valid.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      function a()
3.      {
4.          function b()
5.          {
6.              echo 'I am b';
7.          }
8.          echo 'I am a';
9.      }
10.     a();
11.     a();
12.     ?>
```

- a) I am a
- b) I am bI am a
- c) Error
- d) I am a Error

Answer: a

Explanation: The output will be “I am a” as we are calling a(); so the statement outside the block of function b() will be called.

7. What will be the output of the following PHP code?

```
1.      <?php
2.      function a()
3.      {
4.          function b()
5.          {
```

```

6.         echo 'I am b';
7.     }
8.         echo 'I am a';
9.     }
10.    b();
11.    a();
12.    ?>

```

- a) I am b
- b) I am bI am a
- c) Error
- d) I am a Error

Answer: c

Explanation: The output will be Fatal error: Call to undefined function b(). You cannot call a function which is inside a function without calling the outside function first. It should be a(); then b();

8. What will be the output of the following PHP code?

```

1.    <?php
2.        $op2 = "blabla";
3.        function foo($op1)
4.        {
5.            echo $op1;
6.            echo $op2;
7.        }
8.        foo("hello");
9.    ?>

```

- a) helloblabla
- b) Error
- c) hello
- d) helloblablablabla

Answer: c

Explanation: If u want to put some variables in function that was not passed by it, you must use “global”. Inside the function type global \$op2.

9. A function in PHP which starts with __ (double underscore) is known as _____

- a) Magic Function
- b) Inbuilt Function
- c) Default Function
- d) User Defined Function

Answer: a

Explanation: PHP functions that start with a double underscore – a “__” – are called magic functions in PHP. They are functions that are always defined inside classes, and are not stand-alone functions.

10. What will be the output of the following PHP code?

```
1.      <?php
2.      function foo($msg)
3.      {
4.          echo "$msg";
5.      }
6.      $var1 = "foo";
7.      $var1("will this work");
8.      ?>
```

- a) Error
- b) \$msg
- c) 0
- d) Will this work

Answer: d

Explanation: It is possible to call a function using a variable which stores the function name.

1. Which of the following PHP functions accepts any number of parameters?

- a) func_get_argv()
- b) func_get_args()
- c) get_argv()
- d) get_argc()

Answer: b

Explanation: func_get_args() returns an array of arguments provided. One can use func_get_args() inside the function to parse any number of passed parameters. Here is an example:

```
1. function foo()
2. {
3.     $args = func_get_args();
4.     foreach ($args as $k => $v)
5.     {
6.         echo "arg".($k+1).": $v\n";
7.     }
8. }
9. foo();
10. /* will print nothing */
11.
12. foo("Hello");
13. /* will print Hello */
14.
```

```
15. foo("Hello","World","Bye");  
  
16. /* will print Hello World Bye */
```

2. Which one of the following PHP functions can be used to find files?

- a) glob()
- b) file()
- c) fold()
- d) get_file()

Answer: a

Explanation: The function glob() returns an array of filenames or directories which matches a specified pattern. The function returns an array of files/directories, or it will return FALSE on failure. Here is an example-

```
1. // get all php files AND txt files  
  
2.     $files = glob('*. {php,txt}', GLOB_BRACE);  
  
3.     print_r($files);  
  
4.     /* output looks like:  
  
5.     Array  
  
6.     (  
  
7.         [0] => phptest.php  
  
8.         [1] => pi.php  
  
9.         [2] => post_output.php  
  
10.     .  
  
11.     .  
  
12.     .  
  
13.     )
```

3. Which of the following PHP functions can be used to get the current memory usage?

- a) get_usage()
- b) get_peak_usage()
- c) memory_get_usage()
- d) memory_get_peak_usage()

Answer: c

Explanation: memory_get_usage() returns the amount of memory, in bytes, that's currently being allocated to the PHP script. We can set the parameter 'real_usage' to TRUE to get total memory allocated from system, including unused pages. If it is not set or FALSE then only the used memory is reported. To get the highest amount of memory used at any point, we can use the memory_get_peak_usage() function.

4. Which of the following PHP functions can be used for generating unique ids?

- a) uniqueid()
- b) id()
- c) md5()
- d) mdid()

Answer: a

Explanation: The function uniqueid() is used to generate a unique ID based on the microtime (current time in microseconds). The ID generated from the function uniqueid() is not optimal, as it is based on the system time. To generate an ID which is extremely difficult to predict we can use the md5() function.

5. Which one of the following functions can be used to compress a string?

- a) zip_compress()
- b) zip()
- c) compress()
- d) gzcompress()

Answer: d

Explanation: The function gzcompress() compresses the string using the ZLIB data format. One can achieve upto 50% size reduction using this function. The gzuncompress() function is used to uncompress the string.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      echo chr(52);
3.      ?>
```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: d

Explanation: The chr() function returns a character from the specified ASCII value. We can specify ASCII value in decimal, octal, or hex values. The Octal values are defined as a leading 0, while hex values are defined as a leading 0x. Since the ASCII value of 4 is 52, thus 4 was displayed.

7. What will be the output of the following PHP code?

```
1.      <?php
2.      echo ord ("hi");
3.      ?>
```

- a) 106
- b) 103
- c) 104
- d) 209

Answer: c

Explanation: The ord() function returns the ASCII value of the first character of a string. The ASCII value of h is 104, thus 104 was displayed.

8. What will be the output of the following PHP code?

```
1.      <?php
2.      $str = "Hello World";
3.      echo wordwrap($str,5,"<br>\n");
4.      ?>
```

Answer: b

Explanation: The wordwrap() function wraps a string into new lines when it reaches a specific length.

9. What will be the output of the following PHP code?

```
1.      <?php
```

```
2.      echo ucwords("i love my country");  
3.      ?>
```

- a) I love my country
- b) i love my Country
- c) I love my Country
- d) I Love My Country

Answer: d

Explanation: The ucwords() function converts the first character of each word in a string to uppercase.

10. What will be the output of the following PHP code?

```
1.      <?php  
2.      echo lcfirst("welcome to India");  
3.      ?>
```

- a) welcome to India
- b) welcome to india
- c) Welcome to India
- d) Welcome to india

Answer: a

Explanation: The lcfirst() function converts the first character of a string to lowercase.

1. PHP's numerically indexed array begin with position _____

- a) 1
- b) 2
- c) 0
- d) -1

Answer: c

Explanation: Like all the other programming languages, the first element of an array always starts with '0'.

2. Which of the following are correct ways of creating an array?

```
i) state[0] = "karnataka";  
ii) $state[] = array("karnataka");  
iii) $state[0] = "karnataka";  
iv) $state = array("karnataka");
```

- a) iii) and iv)
- b) ii) and iii)
- c) Only i)
- d) ii), iii) and iv)

Answer: a

Explanation: A variable name should start with \$ symbol which is not present in i) and you need not put the square brackets when you use the array() constructor.

3. What will be the output of the following PHP code?

```
1.      <?php  
2.      $states = array("Karnataka" => array  
3.      ("population" => "11,35,000", "capital" => "Bangalore"),
```

```

4.      "Tamil Nadu" => array( "population" => "17,90,000",
5.      "capital" => "Chennai") );
6.      echo $states["Karnataka"]["population"];
7.      ?>

```

- a) Karnataka 11,35,000
- b) 11,35,000
- c) population 11,35,000
- d) Karnataka population

Answer: b

Explanation: In the following PHP code, the variable states are treated as a multidimensional array and accordingly traverse it to get the value of 'Karnataka's population'.

4. Which of the following PHP function will return true if a variable is an array or false if it is not an array?

- a) this_array()
- b) is_array()
- c) do_array()
- d) in_array()

Answer: b

Explanation: The function is_array() is an inbuilt function in PHP which is used to check whether a variable is an array or not. Its prototype follows: boolean is_array(mixed variable).

5. Which in-built function will add a value to the end of an array?

- a) array_unshift()
- b) into_array()
- c) inend_array()
- d) array_push()

Answer: d

Explanation: array_push adds a value to the end of an array, returning the total count of elements in the array after the new value has been added.

6. What will be the output of the following PHP code?

```

1.      <?php
2.      $state = array ("Karnataka", "Goa", "Tamil Nadu",
3.      "Andhra Pradesh");
4.      echo (array_search ("Tamil Nadu", $state) );
5.      ?>

```

- a) True
- b) 1
- c) False
- d) 2

Answer: d

Explanation: The array_search() function searches an array for a specified value, returning its key if located and FALSE otherwise.

7. What will be the output of the following PHP code?


```
1.      <?php
2.      $fruits = array ("apple", "orange", "banana");
3.      echo (next($fruits));
4.      echo (next($fruits));
5.      ?>
```

- a) orangebanana
- b) appleorange
- c) orangeorange
- d) appleapple

Answer: a

Explanation: The next() function returns the value of the next element in the array. In the first 'next(\$fruits)' call, it will print orange which is next to apple and so on.

8. Which of the following function is used to get the value of the previous element in an array?

- a) last()
- b) before()
- c) prev()
- d) previous()

Answer: c

Explanation: The prev() function returns the previous element in the array.

9. What will be the output of the following PHP code?

```
1.      <?php
2.      $fruits = array ("apple", "orange", array ("pear", "mango"),
3.      "banana");
4.      echo (count($fruits, 1));
5.      ?>
```

- a) 3
- b) 4
- c) 5
- d) 6

Answer: d

Explanation: The function count() will return the number of elements in an array. The parameter 1 counts the array recursively i.e it will count all the elements of multidimensional arrays.

10. Which function returns an array consisting of associative key/value pairs?

- a) count()
- b) array_count()
- c) array_count_values()
- d) count_values()

Answer: c

Explanation: The function array_count_values() will count all the values of an array. It will return an associative array, where the keys will be the original array's values, and the values are the number of occurrences.

1. What will be the output of the following PHP code?

```

1. <?php
2. $cars = array("Volvo", "BMW", "Toyota");
3. echo "I like " . $cars[2] . ", " . $cars[1] . " and " . $cars[0] . ".";
4. ?>

```

- a) I like Volvo, Toyota and BMW
- b) I like Volvo, BMW and Toyota
- c) I like BMW, Volvo and Toyota
- d) I like Toyota, BMW and Volvo

Answer: d

Explanation: The order of elements defined. In the echo statement when we call the elements of array using its index, it will be printed accordingly. As index '0' indicates 'Volvo', '1' for 'BMW' and '2' for Toyota'.

2. What will be the output of the following PHP code?

```

1. <?php
2. $fname = array("Peter", "Ben", "Joe");
3. $age = array("35", "37", "43");
4. $c = array_combine($age, $fname);
5. print_r($c);
6. ?>

```

- a) Array (Peter Ben Joe)
- b) Array ([Peter] => 35 [Ben] => 37 [Joe] => 43)
- c) Array (35 37 43)
- d) Array ([35] => Peter [37] => Ben [43] => Joe)

Answer: d

Explanation: Here "keys" array is \$age and "values" array is \$fname. The function array_combine() will create an array by using the elements from one "keys" array and one "values" array. So when variable c is called, it will print keys and values.

3. What will be the output of the following PHP code?

```

1. <?php
2. $a=array("A","Cat","Dog","A","Dog");
3. $b=array("A","A","Cat","A","Tiger");
4. $c=array_combine($a,$b);
5. print_r(array_count_values($c));
6. ?>

```

- a) Array ([A] => 5 [Cat] => 2 [Dog] => 2 [Tiger] => 1)
- b) Array ([A] => 2 [Cat] => 2 [Dog] => 1 [Tiger] => 1)
- c) Array ([A] => 6 [Cat] => 1 [Dog] => 2 [Tiger] => 1)
- d) Array ([A] => 2 [Tiger] => 1)

Answer: d

Explanation: The function The array_count_values() counts all the values of an array and the The function

array_combine() will create an array by using the elements from one “keys” array and one “values” array.

4. What will be the output of the following PHP code?

```
1. <?php
2. $a1 = array("a" => "red", "b" => "green", "c" => "blue", "d" => "yellow");
3. $a2 = array("e" => "red", "f" => "green", "g" => "blue", "h" => "orange");
4. $a3 = array("i" => "orange");
5. $a4 = array_merge($a2, $a3);
6. $result = array_diff($a1, $a4);
7. print_r($result);
8. ?>
```

- a) Array ([d] => yellow)
- b) Array ([i] => orange)
- c) Array ([h] => orange)
- d) Array ([d] => yellow [h] => orange)

Answer: a

Explanation: The array_diff() function compares the values of two (or more) arrays, and returns the differences. This function compares the values of two (or more) arrays, and return an array that contains the entries from array1 that are not present in other arrays (array2, array3, etc).

5. What will be the output of the following PHP code?

```
1. <?php
2. $a1 = array("red", "green");
3. $a2 = array("blue", "yellow");
4. $a3 = array_merge($a1, $a2);
5. $a4 = array("a", "b", "c", "d");
6. $a = array_combine($a4, $a3);
7. print_r($a);
8. ?>
```

- a) Array ([a] => blue [b] => yellow [c] => red [d] => green)
- b) Array ([0] => blue [1] => yellow [2] => red [3] => green)
- c) Array ([0] => red [1] => green [2] => blue [3] => yellow)
- d) Array ([a] => red [b] => green [c] => blue [d] => yellow)

Answer: d

Explanation: The function array_merge() merges one or more arrays into one array. If in the function array_merge(), two or more array elements have the same key, the last one overrides the others. The function array_combine() will create an array by using the elements from one “keys” array and one “values” array. The program is the basic combined application of array_combine() and array_merge().

6. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = array("a" => "india", "b" => "brazil", "c" => "china");  
3. echo array_shift($a);  
4. echo "<br>";  
5. array_pop($a);  
6. print_r($a);  
7. ?>
```

7. What will be the output of the following PHP code?

```
1. <?php  
2. $a1 = array_fill(1, 4, "hello");  
3. $b1 = array_fill(5, 1, "php");  
4. $a2 = array_merge($a1, $a2);  
5. print_r($a2);  
6. echo "<br>";  
7. print_r($b1);  
8. ?>
```

8. What will be the output of the following PHP code?

```
1. <?php  
2. $names = array("Sam", "Bob", "Jack");  
3. echo $names[0] . "is the brother of " . $names[1] . " and " . $names[1] . ".";  
4. ?>
```

- a) Sam is the brother of Bob and Jack
- b) Sam is the brother of Bob and Bob
- c) Sam is the brother of Jack and Bob
- d) Error

Answer: b

Explanation: Simple definition of array and using it in a string. We have used \$names[1] twice and hence Bob appears twice.

9. What will be the output of the following PHP code?

```
1. <?php  
2. $names = array("Sam", "Bob", "Jack");  
3. echo $names[0]. "is the brother of ".$names[1]. " and ".$names[1]. ".$brother";  
4. ?>
```

- a) Sam is the brother of Bob and Bob) \$brother
- b) Sam is the brother of Bob and Bob)
- c) \$brother
- d) Error

Answer: d

Explanation: \$brother undeclared.

10. What will be the output of the following PHP code?

```
1. <?php
2. $place = array("NYC", "LA", "Paris");
3. array_pop($place);
4. $place1 = array("Paris");
5. $place = array_merge($place, $place1);
6. print_r($place);
7. ?>
```

- a) Array ([0] => LA [1] => Paris [2] => Paris)
- b) Array ([0] => NYC [1] => LA [2] => Paris)
- c) Array ([0] => NYC [1] => LA [2] => Paris [3] => Paris)
- d) Array ([0] => LA [1] => Paris)

Answer: b

Explanation: array_merge() and array_pop() yields that result.

1. What will be the output of the following PHP code?

```
1. <?php
2. $age = array("Harry" => "21", "Ron" => "23", "Malfoy" => "21");
3. array_pop($age);
4. print_r(array_change_key_case($age, CASE_UPPER));
5. ?>
```

- a) Array ([Harry] => 21 [Ron] => 23 [Malfoy] => 21)
- b) Array ([HARRY] => 21 [RON] => 23 [MALFOY] => 21)
- c) Array ([HARRY] => 21 [RON] => 23)
- d) Array ([Harry] => 21 [Ron] => 23)

Answer: c

Explanation: The function array_pop() will delete the last element of an array. So Malfoy => 21 will be deleted and the function array_change_key_case() will change all keys in an array to lowercase or uppercase.

2. What will be the output of the following PHP code?

```
1. <?php
2. $a1 = array("a" => "red", "b" => "green", "c" => "blue", "d" => "yellow");
3. $result = array_flip($a1);
4. print_r($result);
5. ?>
```

- a) Array ([red] => red [green] => green [blue] => blue [yellow] => yellow)
- b) Array ([a] => a [b] => b [c] => c [d] => d)
- c) Array ([red] => a [green] => b [blue] => c [yellow] => d)

d) Array ([a] => red [b] => green [c] => blue [d] => yellow)

Answer: c

Explanation: The function array_flip() flips/exchanges all keys with their associated values in an array. So, in the above program “a” will be flipped with “red”, “b” will be flipped with “green” and so on.

3. What will be the output of the following PHP code?

```
1. <?php
2. $a1 = array("a" => "red", "b" => "green", "c" => "blue", "d" => "yellow");
3. $a2 = array("e" => "red","f" => "green", "g" => "blue");
4. $result = array_intersect($a1, $a2);
5. print_r($result);
6. ?>
```

- a) Array ([a] => red [b] => green [c] => blue)**
- b) Array ([a] => red [b] => green [c] => blue [d] => yellow)**
- c) Array ([e] => red [f] => green [g] => blue)**
- d) Array ([a] => red [b] => green [c] => blue [d] => yellow [e] => red [f] => green [g] => blue)**

Answer: a

Explanation: The function array_intersect() compares the values of two (or more) arrays, and returns the matches. So, in the above program values of a1 and a2 will be compared and the values present in both the arrays will be the returned.

4. What will be the output of the following PHP code?

```
1. <?php
2. $a = array(12, 5, 2);
3. echo(array_product($a));
4. ?>
```

- a) 024**
- b) 120**
- c) 010**
- d) 060**

Answer: b

Explanation: The array_product() function calculates and returns the product of an array.

5. What will be the output of the following PHP code?

```
1. <?php
2. $a = array("a" => "Jaguar", "b" => "Land Rover",
3. "c" => "Audi", "d" => "Maseratti");
4. echo array_search("Audi", $a);
5. ?>
```

- a) a**
- b) b**
- c) c**

d) d

Answer: c

Explanation: The array_search() function searches for the element and returns the key of that element.

6. What will be the output of the following PHP code?

```
1. <?php
2. $city_west = array("NYC", "London");
3. $city_east = array("Mumbai", "Beijing");
4. print_r(array_replace($city_west, $city_east));
5. ?>
```

a) Array ([1] => Mumbai [0] => Beijing)

b) Array ([0] => NYC [1] => London)

c) Array ([1] => NYC [0] => London)

d) Array ([0] => Mumbai [1] => Beijing)

Answer: d

Explanation: The function array_replace() replaces the values of the first array with the values from following arrays. So, in the above program the values of city_west will be replaced with city_east.

7. What will be the output of the following PHP code?

```
1. <?php
2. $people = array("Peter", "Susan", "Edmund", "Lucy");
3. echo pos($people);
4. ?>
```

a) Lucy

b) Peter

c) Susan

d) Edmund

Answer: b

Explanation: The pos() function returns the value of the current element in an array, and since no operation has been done, the current element is the first element.

8. What will be the output of the following PHP code?

```
1. <?php
2. $number = range(0, 5);
3. print_r ($number);
4. ?>
```

a) Array ([0] => 0 [1] => 1 [2] => 2 [3] => 3 [4] => 4 [5] => 5)

b) Array ([0] => 0 [1] => 0 [2] => 0 [3] => 0 [4] => 0 [5] => 0)

c) Array ([0] => 5 [1] => 5 [2] => 5 [3] => 5 [4] => 5 [5] => 5)

d) Array ([0] => 0 [5] => 5)

Answer: a

Explanation: The range() function creates an array containing a range of elements.

9. What will be the output of the following PHP code?

```
1. <?php
2. $array = array("red", "green");
3. array_push($array, "blue", "yellow");
4. print_r($array);
5. ?>
```

- a) Array ([0] => red [1] => green [2] => blue [3] => yellow)
- b) Array ([0] => blue [1] => yellow [2] => red [3] => green)
- c) Array ([0] => red [1] => green)
- d) Array ([0] => blue [1] => yellow)

Answer: a
Explanation: The function `array_push()` inserts one or more elements to the end of an array. So, in the above program blue and yellow will be inserted after previous values.

10. What will be the output of the following PHP code?

```
1. <?php
2. $age = array("Harry" => "21", "Ron" => "19", "Malfoy" => "23");
3. ksort($age);
4. foreach($age as $x => $x_value)
5. {
6.     echo "Key=" . $x . ", Value=" . $x_value;
7.     echo "<br>";
8. }
9. ?>
```

1. What will be the output of the following PHP code?

```
1. <?php
2. $cars = array("Volvo", "BMW", "Toyota");
3. echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
4. ?>
```

- a) I like Volvo BMW and Toyota.
- b) I like Volvo, BMW and Toyota)
- c) I like Volvo, BMW and Toyota.
- d) I like. Volvo,., BMW. and. Toyota)

Answer: b
Explanation: The `array()` function is used to create an array. Each elements are assigned an index as the rule of an array. So, calling `$cars[0]` will print element at index 0 and so on.

2. What will be the output of the following PHP code?

```
1. <?php
```



```

2. $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

3. print_r(array_change_key_case($age, CASE_UPPER));

4. ?>

```

- a) Array ([Peter] => 35 [Ben] => 37 [Joe] => 43)
- b) Array ([peter] => 35 [ben] => 37 [joe] => 43)
- c) Array ([PETER] => 35 [BEN] => 37 [JOE] => 43)
- d) Array ([PeTeR] => 35 [BeN] => 37 [Joe] => 43)

Answer: c

Explanation: The array_change_key_case() function changes all keys in an array to lowercase or uppercase.

3. What will be the output of the following PHP code?

```

1. <?php

2. $cars = array("Volvo", "BMW", "Toyota", "Honda", "Mercedes", "Opel");

3. print_r(array_chunk($cars, 2));

4. ?>

```

- a) Array ([0] => Array ([1] => Volvo [2] => BMW) [1] => Array ([1] => Toyota [2] => Honda) [2] => Array ([1] => Mercedes [2] => Opel))
- b) Array ([1] => Array ([1] => Volvo [2] => BMW) [2] => Array ([1] => Toyota [2] => Honda) [3] => Array ([1] => Mercedes [2] => Opel))
- c) Array ([0] => Array ([0] => Volvo [1] => Volvo) [1] => Array ([0] => BMW [1] => BMW) [2] => Array ([0] => Toyota [1] => Toyota))
- d) Array ([0] => Array ([0] => Volvo [1] => BMW) [1] => Array ([0] => Toyota [1] => Honda) [2] => Array ([0] => Mercedes [1] => Opel))

Answer: d

Explanation: The array_chunk() function splits an array into chunks of new arrays.

4. What will be the output of the following PHP code?

```

1. <?php

2. $fname = array("Peter", "Ben", "Joe");

3. $age = array("35", "37", "43");

4. $c = array_combine($fname, $age);

5. print_r($c);

6. ?>

```

- a) Array (Peter Ben Joe)
- b) Array ([Peter] => 35 [Ben] => 37 [Joe] => 43)
- c) Array (35 37 43)
- d) Array ("[Peter] => 35" "[Ben] => 37" "[Joe] => 43")

Answer: b

Explanation: The array_combine() function creates an array by using the elements from one "keys" array and one "values" array.

5. What will be the output of the following PHP code?

```

1. <?php

```

```
2. $a = array("A", "Cat", "Dog", "A", "Dog");

3. print_r(array_count_values($a));

4. ?>
```

- a) Array ([A] => 2 [Cat] => 1 [Dog] => 2)
- b) Array ([A] => 2 [Cat] => 2 [Dog] => 1)
- c) Array ([A] => 1 [Cat] => 1 [Dog] => 2)
- d) Array ([A] => 2 [Cat] => 1 [Dog] => 1)

Answer: a

Explanation: The array_count_values() function counts all the values of an array.

6. What will be the output of the following PHP code?

```
1. <?php

2. $a1 = array("a"=>"red", "b"=>"green", "c"=>"blue", "d"=>"yellow");

3. $a2 = array("e"=>"red", "f"=>"green", "g"=>"blue");

4. $result = array_diff($a1, $a2);

5. print_r($result);

6. ?>
```

- a) Array ([d] => yellow)
- b) Array ([c] => blue)
- c) Array ([a] => red)
- d) Array ([e] => yellow)

Answer: a

Explanation: The array_diff() function compares the values of two (or more) arrays, and returns the differences.

7. What will be the output of the following PHP code?

```
1. <?php

2. $a1 = array_fill(3, 4, "blue");

3. $b1 = array_fill(0, 1, "red");

4. print_r($a1);

5. echo "<br>";

6. print_r($b1);

7. ?>
```

8. What will be the output of the following PHP code?

```
1. <?php

2. $a1 = array("red", "green");

3. $a2 = array("blue", "yellow");

4. print_r(array_merge($a1, $a2));

5. ?>
```

- a) Array ([0] => red [1] => green)
- b) Array ([0] => blue [1] => yellow [2] => red [3] => green)
- c) Array ([0] => red [1] => green [2] => blue [3] => yellow)
- d) Array ([0] => blue [1] => yellow)

Answer: c

Explanation: The array_merge() function merges one or more arrays into one array.

9. What will be the output of the following PHP code?

```
1. <?php
2. $a = array("a"=>"red", "b"=>"green", "c"=>"blue");
3. echo array_shift($a);
4. print_r ($a);
5. ?>
```

- a) green
- b) red
- c) redArray([c] => green [c] => blue)
- d) redArray([b] => green [c] => blue)

Answer: d

Explanation: The array_shift() function removes the first element from an array, and returns the value of the removed element.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a = array("red", "green", "blue");
3. array_pop($a);
4. print_r($a);
5. ?>
```

- a) Array ([0] => red [1] => green)
- b) Array ([0] => green [1] => blue)
- c) Array ([0] => red [1] => blue)
- d) Array ([0] => blue [1] => blue)

Answer: a

Explanation: The array_pop() function deletes the last element of an array.

1. What will be the output of the following PHP code?

```
1. <?php
2. $fruits = array ("mango", "apple", "pear", "peach");
3. $fruits = array_flip($fruits);
4. echo ($fruits[0]);
5. ?>
```

- a) mango

- b) error
- c) peach
- d) 0

Answer: b

Explanation: As we are flipping the values, \$fruits[“mango”] = 0, \$fruits[“apple”] = 1 and so on.

2. Which of the functions is used to sort an array in descending order?

- a) sort()
- b) asort()
- c) rsort()
- d) dsort()

Answer: c

Explanation: The function sort() will sort the arrays in ascending order, the function rsort() will sort arrays in descending order. While the function asort() will sort associative arrays in ascending order, according to the value.

3. What will be the output of the following PHP code?

```
1.      <?php
2.      $fruits = array ("mango", "apple", "peach", "pear");
3.      $fruits = asort ($fruits);
4.      print_r ($fruits);
5.      ?>
```

- a) Array ([1] => apple [0] => mango [2] => peach [3] => pear)
- b) Array ([0] => apple [1] => mango [2] => peach [3] => pear)
- c) Error
- d) Array ([1] => apple [0] => mango [3] => peach [2] => pear)

Answer: c

Explanation: The program will give an error i.e. uncaught error: call to undefined function print_r(). As the correct function is print_r().

4. What will be the output of the following PHP code?

```
1.      <?php
2.      $arr = array ("picture1.JPG", "picture2.jpg",
3.      "Picture10.jpg", "picture20.jpg");
4.      sort($arr);
5.      print_r($arr);
6.      ?>
```

- a) Array ([0] => picture1.JPG [1] => Picture10.jpg [2] => picture2.jpg [3] => picture20.jpg)
- b) Array ([0] => picture1.JPG [1] => picture2.jpg [2] => Picture10.jpg [3] => picture20.jpg)
- c) Array ([0] => Picture10.jpg [1] => picture1.JPG [2] => picture2.jpg [3] => picture20.jpg)
- d) Array ([0] => Picture10.jpg [1] => picture1.JPG [2] => picture20.jpg [3] => picture2.jpg)

Answer: c

Explanation: The function sort() in PHP sorts an indexed array in ascending order. While sorting, each character is compared with the others and sorted using ASCII values.

5. Which function should we use to sort the array in natural order?

- a) `dsort()`
- b) `casesort()`
- c) `natcasesort()`
- d) `naturalsort()`

Answer: c

Explanation: The function `natcasesort()` in PHP sorts an array by using a “natural order” algorithm. All the values keep their original keys. Eg: In a natural algorithm, as the number 2 is less than the number 10. But in computer sorting, 10 is less than 2, because the first number in “10” is less than 2. The function is case-insensitive.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      $face = array ("A", "J", "Q", "K");
3.      $number = array ("2","3","4", "5", "6", "7", "8", "9", "10");
4.      $cards = array_merge ($face, $number);
5.      print_r ($cards);
6.      ?>
```

- a) Array ([0] => A [1] => J [2] => Q [3] => K [4] => 2 [5] => 3 [6] => 4 [7] => 5 [8] => 6 [9] => 7 [10] => 8 [11] => 9 [12] => 10)
- b) Array ([0] => A [1] => 2 [2] => J [3] => 3 [4] => Q [5] => 4 [6] => K [7] => 5 [8] => 6 [9] => 7 [10] => 8 [11] => 9 [12] => 10)
- c) Error
- d) Array ([0] => 2 [1] => 3 [2] => 4 [3] => 5 [4] => 6 [5] => 7 [6] => 8 [7] => 9 [8] => 10 [9] => A [10] => J [11] => Q [12] => K)

Answer: a

Explanation: The `array_merge()` function merges one or more arrays into one array. The resulting array will begin with the first input array parameter, appending each subsequent array parameter in the order of appearance.

7. What will be the output of the following PHP code?

```
1.      <?php
2.      $fruits = array ("apple", "mango", "peach", "pear",
3.      "orange");
4.      $subset = array_slice ($fruits, 2);
5.      print_r ($subset);
6.      ?>
```

- a) Array ([0] => peach)
- b) Array ([0] => apple [1] => mango [2] => peach)
- c) Array ([0] => apple [1] => mango)
- d) Array ([0] => peach [1] => pear [2] => orange)

Answer: d

Explanation: The `array_slice()` function returns a section of an array based on a starting and ending offset value.

8. What will be the output of the following PHP code?

```

1.      <?php
2.      $fruits = array ("apple", "mango", "peach", "pear",
3.      "orange");
4.      $subset = array_splice ($fruits, 2);
5.      print_r ($fruits);
6.      ?>

```

- a) Error
- b) Array ([0] => apple [1] => mango [2] => peach)
- c) Array ([0] => apple [1] => mango)
- d) Array ([0] => pear [1] => orange)

Answer: c

Explanation: The `array_splice()` function removes all elements of an array found within a specified range.

9. What will be the output of the following PHP code?

```

1.      <?php
2.      $number = array ("4", "hello", 2);
3.      echo (array_sum ($number));
4.      ?>

```

- a) 4hello2
- b) 4
- c) 2
- d) 6

Answer: d

Explanation: The `array_sum()` function add all the values of the input array together, returning the final sum. If a string datatype is found, it'll be ignored.

10. What will be the output of the following PHP code?

```

1.      <?php
2.      $array1 = array ("KA", "LA", "CA", "MA", "TA");
3.      $array2 = array ("KA", "IA", "CA", "GA", "TA");
4.      $inter = array_intersect ($array1, $array2);
5.      print_r ($inter);
6.      ?>

```

- a) Array ([0] => KA [1] => LA [2] => CA [3] => MA [4] => TA [5] => IA [6] => GA)
- b) Array ([0] => KA [2] => CA [4] => TA)
- c) Array ([1] => IA [3] => GA)
- d) Array ([1] => LA [3] => MA)

Answer: b

Explanation: The `array_intersect()` function returns a key preserved array consisting only of those values present in the first array that are also present in each of the other input arrays.

1. The practice of separating the user from the true inner workings of an application through well-known interfaces is known as _____

- a) Polymorphism**
- b) Inheritance**
- c) Encapsulation**
- d) Abstraction**

Answer: c

Explanation: In object-oriented PHP encapsulation is a concept of wrapping up or binding up the data members and methods in a single module.

2. Which of the following term originates from the Greek language that means “having multiple forms,” defines OOP’s ability to redefine, a class’s characteristics?

- a) Abstraction**
- b) Polymorphism**
- c) Inheritance**
- d) Differential**

Answer: b

Explanation: The word polymorphism is derived from Greek word poly which means “many” and morphism which means the property which helps us to assign more than one property.

3. The practice of creating objects based on predefined classes is often referred to as _____

- a) class creation**
- b) object creation**
- c) object instantiation**
- d) class instantiation**

Answer: d

Explanation: In object-oriented programming, classes are the blueprints of php objects. Classes do not actually become objects until instantiation is done. When someone instantiates a class, it creates an instance of it, thus creating the object. In other words, instantiation is the process of creating an instance of an object in memory.

4. Which one of the following property scopes is not supported by PHP?

- a) friendly**
- b) final**
- c) public**
- d) static**

Answer: a

Explanation: PHP supports five class property scopes: public, private, protected, final and static.

5. Which one of the following can be used to instantiate an object in PHP assuming class name to be Foo?

- a) \$obj = new \$foo;**
- b) \$obj = new foo;**
- c) \$obj = new foo ();**
- d) obj = new foo ();**

Answer: c

Explanation: To create a new object in PHP we can use the new statement to instantiate a class.

6. Which one of the following is the right way to define a constant?

- a) constant PI = “3.1415”;**
- b) const \$PI = “3.1415”;**
- c) constant PI = ‘3.1415’;**
- d) const PI = ‘3.1415’;**

Answer: d

Explanation: Class constants are created like: `const NAME = 'VALUE';`

7. Which one of the following is the right way to call a class constant, given that the class is `mathFunction`?

- a) `echo PI;`**
- b) `echo mathFunction->PI;`**
- c) `echo mathFunction::PI;`**
- d) `echo mathFunction=PI;`**

Answer: c

Explanation: The Scope Resolution Operator “::” is a token that allows access to static, constant, and overridden properties or methods of a class.

8. Which one of the following is the right way to invoke a method?

- a) `$object->methodName();`**
- b) `object->methodName();`**
- c) `object::methodName();`**
- d) `$object::methodName();`**

Answer: a

Explanation: “->” is a dynamic class method invocation in PHP.

- a) Only ii)**
- b) Only iv)**
- c) i) and ii)**
- d) iii) and iv)**

Answer: c

Explanation: In case of public methods, you can forgo explicitly declaring the scope and just declare the method like you would a function.

- a) Only ii)**
- b) Only iv)**
- c) ii) and iv)**
- d) Only i)**

Answer: a

Explanation: PHP supports six method scopes: public, private, final, static, protected and abstract. But it does not support friendly.

1. Which method scope prevents a method from being overridden by a subclass?

- a) Abstract**
- b) Protected**
- c) Final**
- d) Static**

Answer: c

Explanation: When we declare a method as final then it is not possible to override that method. Methods should not be overridden due to some security or any other reasons.

- a) ii)**
- b) ii) and iii)**
- c) i), ii), iii) and iv)**
- d) ii), iii) and iv)**

Answer: c

Explanation: If a class name and the function name is similar then the function is known as constructor. Constructor is automatically called when an object will be initialized. Constructors can accept parameters. Constructors can call class methods or other functions. Class constructors can call on other constructors.

3. PHP recognizes constructors by the name _____

- a) classname()
- b) _construct()
- c) function _construct()
- d) function __construct()

Answer: d

Explanation: A double underscore followed by the construct keyword. Its syntax is function __construct ([argument1, argument2,]) { Class Initialization code }.

4. Which version of PHP introduced the instanceof keyword?

- a) PHP 4
- b) PHP 5
- c) PHP 5.3
- d) PHP 6

Answer: b

Explanation: Using instanceof keyword we can determine whether an object is an instance of a class. \$manager = new Employee() ... if (\$manager instanceof Employee) echo "True";

5. Which one of the following functions is used to determine whether a class exists?

- a) exist()
- b) exist_class()
- c) class_exist()
- d) __exist()

Answer: c

Explanation: The class_exist() function returns true or false according to whether the class exists within the currently executing script content.

6. Which one of the following functions is used to determine object type?

- a) obj_type()
- b) type()
- c) is_a()
- d) is_obj()

Answer: c

Explanation: The is_a() function returns true if object belongs to a class type or if it belongs to a class that is a child of that class. Or else false is returned.

7. Which one of the following keyword is used to inherit our subclass into a superclass?

- a) extends
- b) implements
- c) inherit
- d) include

Answer: a

Explanation: When we extend a class then the subclass will inherit all the public and protected methods from the parent class.

The keyword implements are used with interfaces. With inheritance, we use the keyword extends.

8. In the following PHP code, what is/are the properties?

```
1.      <?php
2.      class Example
3.      {
```

```

4.         public $name;
5.
6.         function Sample()
7.         {
8.             echo "This is an example";
9.         }
10.    ?>

```

- a) echo “This is an example”;
- b) public \$name;
- c) class Example
- d) function sample()

Answer: b

Explanation: Above code is an example of ‘classes’ in PHP. Classes are the blueprints of objects. Classes are the programmer-defined data type, which includes the local methods and the local variables. Class is a collection of objects which has properties and behaviour.

9. Which keyword is used to refer to properties or methods within the class itself?

- a) private
- b) public
- c) protected
- d) \$this

Answer: d

Explanation: In PHP, the self and ‘this’ keyword are used to refer the class members within the scope of a class itself. The class members can be either variables or functions.

10. Which keyword allows class members (methods and properties) to be used without needing to instantiate a new instance of the class?

- a) protected
- b) final
- c) static
- d) private

Answer: c

Explanation: Sometimes it is very handy to access the methods and properties in terms of a class rather than an object. But this can be done through static keyword. Any method declared as ‘static’ can be accessed without the creation of an object.

- a) i)
- b) ii)
- c) i) and ii)
- d) iii) and iv)

Answer: c

Explanation: The advanced OOP features are: Object cloning, Inheritance, Interfaces, Abstract classes, and Namespaces.

2. Which version of PHP introduced the advanced concepts of OOP?

- a) PHP 4
- b) PHP 5
- c) PHP 5.3
- d) PHP 6

Answer: b

Explanation: Advanced concepts of OOP were introduced in PHP version 5.

3. Which one of the following is the right way to clone an object?

- a) `_clone(targetObject);`
- b) `destinationObject = clone targetObject;`
- c) `destinationObject = _clone(targetObject);`
- d) `destinationObject = clone(targetObject);`

Answer: b

Explanation: You can clone an object by prefacing it with the clone keyword. A copy of an object is created by using the clone keyword. `$copy_of_object = clone $object;`

- a) Only i)
- b) ii), iii) and iv)
- c) Only iii)
- d) ii) and iv)

Answer: d

Explanation: The class whose properties are inherited by child class is called Base Class or Parent class.

Answer: a

Explanation: The abstract classes are the classes in which at least one method need to be abstract. Abstract classes in PHP are declared with the help of abstract keyword.

6. If one intends to create a model that will be assumed by a number of closely related objects, which class must be used?

- a) Normal class
- b) Static class
- c) Abstract class
- d) Interface

Answer: c

Explanation: The abstract classes are the classes in which at least one method need to be abstract. Abstract classes in PHP are declared with the help of keyword abstract. The use of class abstract are that all base classes implementing abstract class should give implementation of abstract methods declared in parent class.

7. If your object must inherit behavior from a number of sources you must use a/an

- a) Interface
- b) Object
- c) Abstract class
- d) Static class

Answer: a

Explanation: An interface in PHP consists of methods that have no implementations, i.e. the interface methods are abstract methods. The methods in the interfaces must have public visibility scope. The interfaces are different from classes as the class can inherit from one class only whereas the class can implement one or more interfaces.

8. Which method is used to tweak an object’s cloning behavior?

- a) `clone()`
- b) `__clone()`
- c) `_clone`
- d) `object_clone()`

Answer: b

Explanation: A copy of an object is created by using the clone keyword, which calls the object’s `__clone()` method.

9. Which feature allows us to call more than one method or function of the class in single instruction?

- a) **Typecasting**
- b) **Method Including**
- c) **Method adding**
- d) **Method chaining**

Answer: d

Explanation: When many methods are called in a single instruction in PHP, it is called method chaining. Following is a basic example of method chaining in php:

```
$a = new Order();  
$a->CreateOrder()->sendOrderEmail()->createShipment();
```

10. Which magic method is used to implement overloading in PHP?

- a) **__call**
- b) **__invoke**
- c) **__wakeup**
- d) **__unset**

Answer: a

Explanation: When a class implements __call(), then an object of that class is called with a method that doesn't exist, __call() is called instead.

1. How many error levels are available in PHP?

- a) **14**
- b) **15**
- c) **16**
- d) **17**

Answer: c

Explanation: Whenever the PHP engine encounters any problem that prevents a script from running properly it generates an error message. There are sixteen error levels and each level is represented by an integer value and an associated constant.

2. What is the description of Error level E_ERROR?

- a) **Fatal run-time error**
- b) **Near-fatal error**
- c) **Compile-time error**
- d) **Fatal Compile-time error**

Answer: a

Explanation: E_ERROR is a fatal run-time error, that can't be recovered from and the execution of the script is stopped immediately.

3. Which version of PHP introduced E_STRICT Error level?

- a) **PHP 4**
- b) **PHP 5**
- c) **PHP 5.2**
- d) **PHP 5.3**

Answer: b

Explanation: E_STRICT is PHP version portability suggestions. It is not strictly an error, but it is triggered whenever PHP encounters code that could lead to problems or forward incompatibilities.

4. Which character does the error_reporting directive use to represent the logical operator NOT?

- a) **/**
- b) **!**
- c) **~**
- d) **^**

Answer: c

Explanation: The twiddle (~) character is used to represent the logical operator NOT.

5. Say you want to report error concerned about fatal run-time, fatal compile-time error and core error which statement would you use?

- a) `error_reporting = E_ALL`**
- b) `error_reporting = E_ERROR | E_PARSE | E_CORE_ERROR`**
- c) `error_reporting = E_ERROR | E_COMPILE_WARNING | E_CORE_ERROR`**
- d) `error_reporting = E_ERROR | E_COMPILE_ERROR | E_CORE_ERROR`**

Answer: d

Explanation: E_ERROR is a fatal run-time error, that can't be recovered from. E_COMPILE_ERROR is a fatal error that occurs while the script was being compiled. And E_CORE_ERROR is a fatal error that occurs during the PHP's engine initial startup.

6. Which version introduced the function `error_get_last()`?

- a) PHP 4**
- b) PHP 5**
- c) PHP 5.2**
- d) PHP 5.3**

Answer: c

Explanation: This function returns an associative array consisting of the type, message, file, and line of the last occurring error.

7. Which of the following statements causes PHP to disregard repeated error messages that occur within the same file and on the same line?

- a) `ignore_repeated_errors`**
- b) `ignore_repeat_error`**
- c) `repeatedly_ignore_error`**
- d) `repeated_error_ignore`**

Answer: a

Explanation: `ignore_repeated_errors` will not log repeated messages. The repeated errors must occur in the same file on the same line unless `ignore_repeated_source` is set to true.

8. Which function initializes the constants necessary for using the `openlog()`, `clndelog()`, and `syslog()` functions?

- a) `define_variable()`**
- b) `define_log_variable()`**
- c) `log_variable()`**
- d) `define_syslog_variable()`**

Answer: d

Explanation: If you're running PHP version 5.2.X or older, you need to execute this function before using any of the following logging functions.

9. Which logging option's description is if an error occurs when writing to the syslog, send output to the system console?

- a) `LOG_CONS`**
- b) `LOG_NDELAY`**
- c) `LOG_ODELAY`**
- d) `LOG_PERROR`**

Answer: a

Explanation: If there is an error while sending data to the system logger, `LOG_CONS` will write directly to the system console.

10. Which function is responsible for sending a custom message to the system log?

- a) `systemlog()`
- b) `syslog()`
- c) `log_system()`
- d) `sys_log()`

Answer: b

Explanation: The function `syslog()` generates a log message that will be distributed by the system logger.

1. Which version of PHP was added with Exception handling?

- a) PHP 4
- b) PHP 5
- c) PHP 5.3
- d) PHP 6

Answer: b

Explanation: Exception handling was added to PHP with the version 5 release, and further enhanced with version 5.3.

2. How many methods are available for the exception class?

- a) 5
- b) 6
- c) 7
- d) 8

Answer: c

Explanation: The seven methods are: `getCode()`, `getFile()`, `getLine()`, `getMessage()`, `getPrevious()`, `getTrace()`, `getTraceAsString()`.

3. Which version added the method `getPrevious()`?

- a) PHP 4
- b) PHP 5
- c) PHP 5.1
- d) PHP 5.3

Answer: d

Explanation: The function `getPrevious()` returns previous exception.

4. Which of the following statements invoke the exception class?

- a) `throws new Exception();`
- b) `throw new Exception();`
- c) `new Exception();`
- d) `new throws Exception();`

Answer: b

Explanation: `throw new Exception();` trigger an exception and each “throw” must have at least one “catch”.

5. Which one of the following is the right description for the method `getMessage()`?

- a) Returns the message if it is passed to the constructor
- b) Returns the message if it is passed to the class
- c) Returns the message if it is passed to the file
- d) Returns the message if it is passed to the object

Answer: a

Explanation: The function `getMessage()` gets the exception message. It returns the message if it is passed to the constructor.

6. You can extend the exception base class, but you cannot override any of the preceding methods because they are declared as _____

- a) protected

- b) final
- c) static
- d) private

Answer: b

Explanation: Marking a method as final prevents it from being overridden by a subclass.

7. What does SPL stand for?

- a) Standard PHP Library
- b) Source PHP Library
- c) Standard PHP List
- d) Source PHP List

Answer: a

Explanation: The standard PHP library(SPL) extends PHP by offering ready-made solutions to commonplace tasks such as file access, iteration of various sorts etc.

8. How many predefined exceptions does SPL provide access to?

- a) 13
- b) 14
- c) 15
- d) 16

Answer: a

Explanation: It provides 13 exceptions: BadFunctionCallException, BadMethodCallException, DomainException, InvalidArgumentException, LengthException, LogicException, OutOfBoundsException, OutOfRangeException, OverflowException, RangeException, RuntimeException, UnderflowException, UnexpectedValueException.

- a) Only ii)
- b) Only iii)
- c) i), ii), iii) and iv)
- d) Only iv)

Answer: c

Explanation: There are total 13 types of predefined exceptions SPL provide access to.

BadFunctionCallException, BadMethodCallExceptio, LogicException, DomainException are also included in the exceptions.

- a) i)
- b) i) and iii)
- c) i) and ii)
- d) i), ii), iii) and iv)

Answer: d

Explanation: The exception is thrown if a value is not a valid key. This represents errors that cannot be detected at compile time. OutOfBoundException, OutOfRangeException, OverflowException, UnderflowException are valid exceptions in PHP.

- a) Only ii)
- b) ii) and iii)
- c) Only iii)
- d) i), ii), iii) and iv)

Answer: d

Explanation: The web applications receives external input. External input/data can be: User input from a form, Cookies, Web services data, Server variables and, Database query results.

2. How many types of filtering are present in PHP?

- a) 3
- b) 2
- c) 4
- d) None

Answer: b

Explanation: There are two main types of filtering: validation and sanitization.

3. Which one of the following filter is used to filter several variables with the same or different filters?

- a) `filter_var_array()`
- b) `filter_var()`
- c) `filter_input`
- d) `filter_input_array`

Answer: a

Explanation: The function `filter_var_array()` can get multiple variables and it optionally filters them. The function is useful for filtering many values without calling `filter_var()`.

4. What will be the output of the following PHP code?

```
1.      <?php
2.      $num = "123";
3.      if (!filter_var($num, FILTER_VALIDATE_INT))
4.          echo("Integer is not valid");
5.      else
6.          echo("Integer is valid");
7.      ?>
```

- a) No output is returned
- b) Integer is not valid
- c) Integer is valid
- d) Error

Answer: c

Explanation: The function `filter_var()` can validate and sanitize data. This function filters a single variable with a specified filter.

5. Which one of the following does not describe a validating filter?

- a) Are used to allow or disallow specific characters in a string
- b) Are used to validate user input
- c) Strict format rules
- d) Returns the expected type on success or FALSE on failure

Answer: a

Explanation: Validate filter are used to validate user input, it have strict format rules and it returns the expected type on success or FALSE on failure but 'are used to allow or disallow specific characters in a string' describes sanitizing filters.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      $var=300;
3.      $int_options = array("options"=>array ("min_range"=>0, "max_range"=>256));
```



```

4.         if (!filter_var($var, FILTER_VALIDATE_INT, $int_options))
5.             echo("Integer is not valid");
6.         else
7.             echo("Integer is valid");
8.         ?>

```

- a) No output is returned
- b) Integer is not valid
- c) Integer is valid
- d) Error

Answer: b

Explanation: Since the integer is “300” it is not in the specified range, and the output of the code above will be: “Integer is not valid”.

7. If the input variable is a string like this “http://www.saåånfoøøundry.com/”, the \$url variable after the sanitizing will look like?

- a) http://www.saåånfoøøundry.com/
- b) http://www.saaanfoooundry.com/
- c) http://www.saånfoøundry.com/
- d) https://www.sanfoundry.com/

Answer: d

Explanation: Sanitize is nothing but take away invalid characters so therefore the invalid characters like å and ø will be removed.

8. Which one of the following filter checks if the variable of the specified type exists?

- a) filter_has_var
- b) filter_var
- c) filter_id
- d) filter_var_array

Answer: a

Explanation: The filter filter_has_var checks if the variable of the specified type exists. Whereas the function filter_id() returns filter ID of a specified filter name. The function filter_var() can validate and sanitize data. The function filter_var_array() can get multiple variables and it optionally filters them.

9. What will be the output of the following PHP code?

```

1.     <?php
2.         $value = 'car';
3.         $result = filter_var($value, FILTER_VALIDATE_BOOLEAN, FILTER_NULL_ON_FAILURE);
4.         ?>

```

- a) FALSE
- b) TRUE
- c) NULL
- d) ERROR

Answer: c

Explanation: There is an undocumented filter flag for FILTER_VALIDATE_BOOLEAN. The documentation implies that it will return NULL if the value doesn't match the allowed true/false values. However this doesn't happen unless you give it the FILTER_NULL_ON_FAILURE flag.

10. What will be the output of the following PHP code?

```
1.      <?php
2.      function convertSpace($string)
3.      {
4.          return str_replace("_", " ", $string);
5.      }
6.      $string = "Peter_is_a_great_guy!";
7.      echo filter_var($string, FILTER_CALLBACK, array("options"=>"convertSpace"));
8.      ?>
```

- a) Peter_is_a_great_guy!**
- b) Peterisagreatguy!**
- c) Peter is a great guy!**
- d) Error**

Answer: c

Explanation: The code above converts all “_” to white spaces. Call the filter_var() function with the FILTER_CALLBACK filter and an array containing our function.

- a) i) and ii)**
- b) ii) and iv)**
- c) i) and iv)**
- d) ii) and iii)**

Answer: c

Explanation: None.

2. Which one of the following regular expression matches any string containing zero or one p?

- a) p+**
- b) p***
- c) P?**
- d) p#**

Answer: c

Explanation: None.

3. [:alpha:] can also be specified as _____

- a) [A-Za-z0-9]**
- b) [A-za-z]**
- c) [A-z]**
- d) [a-z]**

Answer: b

Explanation:[:alpha:] is nothing but Lowercase and uppercase alphabetical characters.

4. How many functions does PHP offer for searching strings using POSIX style regular expression?

- a) 7**
- b) 8**
- c) 9**
- d) 10**

Answer: a

Explanation: `ereg()`, `ereg_replace()`, `eregi()`, `eregi_replace()`, `split()`, `spliti()`, and `sql_regcase()` are the functions offered.

5. What will be the output of the following PHP code?

```
1.      <?php
2.      $username = "jasoN";
3.      if (ereg("([a-z])",$username))
4.          echo "Username must be all lowercase!";
5.      else
6.          echo "Username is all lowercase!";
7.      ?>
```

- a) Error
- b) Username must be all lowercase!**
- c) Username is all lowercase!
- d) No Output is returned

Answer: b
Explanation: Because the provided username is not all lowercase, `ereg()` will not return FALSE (instead returning the length of the matched string, which PHP will treat as TRUE), causing the message to output.

6. POSIX implementation was deprecated in which version of PHP?

- a) PHP 4
- b) PHP 5**
- c) PHP 5.2
- d) PHP 5.3

Answer: d
Explanation: None.

7. POSIX stands for _____

- a) Portable Operating System Interface for Unix**
- b) Portable Operating System Interface for Linux
- c) Portative Operating System Interface for Unix
- d) Portative Operating System Interface for Linux

Answer: a
Explanation: None.

8. What will be the output of the following PHP code?

```
1.      <?php
2.      $text = "this is\tsome text that\nwe might like to parse.";
3.      print_r(split("\n\t",$text));
4.      ?>
```

- a) this is some text that we might like to parse.**
- b) Array ([0] => some text that [1] => we might like to parse.)
- c) Array ([0] => this is [1] => some text that [2] => we might like to parse.)
- d) [0] => this is [1] => some text that [2] => we might like to parse.**

Answer: d

Explanation: The split() function divides a string into various elements, with the boundaries of each element based on the occurrence of a defined pattern within the string.

- a) Only i)
- b) ii) and iii)
- c) i), iii) and iv)
- d) i) and iv)

Answer: b
Explanation: This matches f followed by two to four occurrences of o.

- a) Only i)
- b) i) and iii)
- c) ii), iii) and iv)
- d) ii) and iv)

Answer: a
Explanation: /A, /b and /B are metacharacters. \A: Matches only at the beginning of the string. \b: Matches a word boundary. \B: Matches anything but a word boundary.

- 1. How many functions does PHP offer for searching and modifying strings using Perl-compatible regular expressions.**
- a) 7
 - b) 8
 - c) 9
 - d) 10

Answer: b
Explanation: The functions are preg_filter(), preg_grep(), preg_match(), preg_match_all(), preg_quote(), preg_replace(), preg_replace_callback(), and preg_split().

2. What will be the output of the following PHP code?

```
1.      <?php
2.      $foods = array("pasta", "steak", "fish", "potatoes");
3.      $food = preg_grep("/^s/", $foods);
4.      print_r($food);
5.      ?>
```

- a) Array ([0] => pasta [1] => steak [2] => fish [3] => potatoes)
- b) Array ([3] => potatoes)
- c) Array ([1] => steak)
- d) Array ([0] => potatoes)

Answer: c
Explanation: This function is used to search an array for foods beginning with s.

- a) i) and ii)
- b) iii) and iv)
- c) only i)
- d) i), ii), iii) and iv)

Answer: d
Explanation: All of the functions mentioned above can be used to compare strings in some or the other way.

- 4. Which one of the following functions will convert a string to all uppercase?**
- a) strtoupper()

- b) `uppercase()`
- c) `str_uppercase()`
- d) `struppercase()`

Answer: a

Explanation: Its prototype follows `string strtoupper(string str)`.

5. What will be the output of the following PHP code?

```
1.      <?php
2.      $title = "O'malley wins the heavyweight championship!";
3.      echo ucwords($title);
4.      ?>
```

- a) O'Malley Wins The Heavyweight Championship!
- b) O'malley Wins The Heavyweight Championship!
- c) O'Malley wins the heavyweight championship!
- d) o'malley wins the heavyweight championship!

Answer: a

Explanation: The `ucwords()` function capitalizes the first letter of each word in a string. Its prototype follows: `string ucwords(string str)`.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      echo str_pad("Salad", 5)." is good.";
3.      ?>
```

- a) SaladSaladSaladSaladSalad is good
- b) is good SaladSaladSaladSaladSalad
- c) is good Salad
- d) Salad is good

Answer: d

Explanation: The `str_pad()` function pads a string with a specified number of characters.

7. Which one of the following functions can be used to concatenate array elements to form a single delimited string?

- a) `explode()`
- b) `implode()`
- c) `concat()`
- d) `concatenate()`

Answer: b

Explanation: None.

8. Which one of the following functions finds the last occurrence of a string, returning its numerical position?

- a) `strlastpos()`
- b) `strpos()`
- c) `strlast()`
- d) `strrpos()`

Answer: d

Explanation: None.

9. What will be the output of the following PHP code?

```
1. <?php
2. $author = "[email protected]";
3. $author = str_replace("a", "@", $author);
4. echo "Contact the author of this article at $author.";
5. ?>
```

- a) Contact the author of this article at [\[email protected\]](#)@mple.com
- b) [\[email protected\]](#) the @uthor of this @rticle @t [\[email protected\]](#)@[\[email protected\]](#)
- c) Contact the author of this article at [\[email protected\]](#)@[\[email protected\]](#)
- d) Error

Answer: c

Explanation: The `str_replace()` function case sensitively replaces all instances of a string with another.

10. What will be the output of the following PHP code?

```
1. <?php
2. $url = "[email protected]";
3. echo ltrim(strstr($url, "@"), "@");
4. ?>
```

- a) [\[email protected\]](#)
- b) nachiketh
- c) [\[email protected\]](#)
- d) example.com

Answer: d

Explanation: The `strstr()` function returns the remainder of a string beginning with the first occurrence of a predefined string.

1. The `filesize()` function returns the file size in _____

- a) bits
- b) bytes
- c) kilobytes
- d) gigabytes

Answer: b

Explanation: The function `filesize()` returns the size of the specified file and it returns the file size in bytes on success or `FALSE` on failure.

2. Which one of the following PHP function is used to determine a file's last access time?

- a) `fileltime()`
- b) `filectime()`
- c) `fileatime()`
- d) `filetime()`

Answer: c

Explanation: The `fileatime()` function returns a file's last access time in Unix timestamp format or `FALSE` on error.

3. Which one of the following function is capable of reading a file into an array?

- a) `file()`

- b) **arrfile()**
- c) **arr_file()**
- d) **file_arr()**

Answer: a

Explanation: The function file() will read the entire file into an array.

4. Which one of the following function is capable of reading a file into a string variable?

- a) **file_contents()**
- b) **file_get_contents()**
- c) **file_content()**
- d) **file_get_content()**

Answer: b

Explanation: The function file_get_contents() reads a file into a string. This is the preferred way to read the contents of a file into a string as it will use memory mapping techniques.

5. Which one of the following function is capable of reading a specific number of characters from a file?

- a) **fgets()**
- b) **fget()**
- c) **fileget()**
- d) **filegets()**

Answer: a

Explanation: The function fgets() will return a line from an open file. This stops returning on a new line, at the specified length, or at EOF, whichever comes first. Its prototype is string fgets(resource handle [, int length]). If the optional length parameter is omitted, 1024 character is assumed.

6. Which one of the following function operates similarly to fgets(), except that it also strips any HTML and PHP tags form the input?

- a) **fgetsh()**
- b) **fgetsp()**
- c) **fgetsa()**
- d) **fgetss()**

Answer: d

Explanation: The function fgetss() returns a line, with HTML and PHP tags removed, from an open file. This function operates similarly to fgets(), except that it also strips any HTML and PHP tags form the input.

7. Which one of the following function outputs the contents of a string variable to the specified resource?

- a) **filewrite()**
- b) **fwrite()**
- c) **filewrites()**
- d) **fwrites()**

Answer: b

Explanation: The function fwrite() writes to an open file. This will stop at the end of the file or when it reaches the specified length, whichever comes first.

8. Which function sets the file filename last-modified and last-accessed times?

- a) **sets()**
- b) **set()**
- c) **touch()**
- d) **touched()**

Answer: c

Explanation: The function touch() will set the access and modification time of the specified file. Syntax is touch(filename, time, atime).

9. Which function is useful when you want to output the executed command result?

- a) out_cmm()
- b) out_system()
- c) cmm()
- d) system()

Answer: d

Explanation: The function system() in PHP is same as the C version of the function as in that it executes the given command and outputs the result. This function also tries to automatically flush the web server's output buffer after each line of output if PHP is running as a server module.

10. Which one of the following function reads a directory into an Array?

- a) scandir()
- b) readdir()
- c) scandirectory()
- d) readdirectory()

Answer: a

Explanation: It returns an array consisting of files and directories found in directory or returns FALSE on error.

1. What will be the output of the following PHP code?

```
1.      <?php
2.      echo (checkdate(4,31,2010) ? 'Valid' : 'Invalid');
3.      ?>
```

- a) TRUE
- b) FALSE
- c) Valid
- d) Invalid

Answer: d

Explanation: The function checkdate() is used to validate a Gregorian date. In the program, April has 30 days and the above date is 31 therefore Invalid is returned.

2. The date() function returns ____ representation of the current date and/or time.

- a) Integer
- b) String
- c) Boolean
- d) Float

Answer: b

Explanation: The function date() is used to format a local date and time, and it will return the formatted date string. The syntax of this function is string date(string format [, int timestamp]).

3. Which one of the following format parameter can be used to identify timezone?

- a) T
- b) N
- c) E
- d) I

Answer: c

Explanation: When the format is E the time zone is identified and returned, for example, America/New York. N denotes ISO-8601 numeric representation of the day of the week, T denotes time zone abbreviation, I denotes whether or not the date is in daylight saving time.

4. If the format is F then which one of the following will be returned?

- a) Complete text representation of month
- b) Day of month, with leading zero
- c) Daylight saving time
- d) Day of month, without zeros

Answer: a

Explanation: F represents a full textual representation of a month, such as January or March. Day of month, with leading zero is represented by D, Daylight saving time by I, Day of month without zeros by j.

5. What will be the output of the following code? (If say date is 22/06/2013.)

```
1.      <?php
2.      echo "Today is ".date("F d, Y");
3.      ?>
```

- a) Today is 22 June, 2013
- b) Today is 22-06-2013
- c) Today is 06-22-2013
- d) Today is June 22, 2013

Answer: d

Explanation: The function date() is used to format a local date and time, and it will return the formatted date string. F is the parameter for complete text representation of month, d for day of month, and Y for 4 digit representation of year.

6. Which one of the following function is useful for producing a timestamp based on a given date and time?

- a) time()
- b) mktime()
- c) mertime()
- d) mtime()

Answer: b

Explanation: The function mktime() returns the Unix timestamp for a date. This function is same as gmmktime() except that the passed parameters represents a date not a GMT date.

7. Which function displays the web page’s most recent modification date?

- a) lastmod()
- b) getlastmod()
- c) last_mod()
- d) get_last_mod()

Answer: b

Explanation: The function getlastmod() gets the time of the last modification of the main script of execution. It returns the value of the page’s last modified header or FALSE in the case of an error.

8. What will be the output of the following PHP code? (If say date is 22/06/2013.)

```
1.      <?php
2.      printf( date("t") );
3.      ?>
```

- a) 30
- b) 22
- c) JUNE
- d) 2013

Answer: a

Explanation: The t parameter is used to determine the number of days in the current month.

9. Suppose you want to calculate the date 45 days from the present date which one of the following statement will you use?

- a) `totime("+45")`
- b) `totime("+45 days")`
- c) `strtotime("+45 days")`
- d) `strtotime("-45 days")`

Answer: c

Explanation: The strtotime() function and GNU date syntax is used to calculating the date x days from the present date.

10. To create an object and set the date to JUNE 22, 2013, which one of the following statement should be executed?

- a) `$date = Date("22 JUNE 2013")`
- b) `$date = new Date("JUNE 22 2013")`
- c) `$date = DateTime("22 JUNE 2013")`
- d) `$date = new DateTime("22 JUNE 2013")`

Answer: d

Explanation: The dateTime() method is class constructor. You can set the date either at the time of instantiation or later by using a variety of mutators.

1. How many methods does the DateTime class have?

- a) 8
- b) 9
- c) 10
- d) 11

Answer: b

Explanation: The DateTime class has nine methods, all of which are public.

2. How many constants does the DateTime class have?

- a) 8
- b) 9
- c) 10
- d) 11

Answer: d

Explanation: The DateTime class has 11 constants, no static properties or methods.

3. Which method is simply an object-oriented version of date()?

- a) `DateTime::format()`
- b) `DateTime::modify()`
- c) `DateTime::setTime()`
- d) `DateTime::setDate()`

Answer: a

Explanation: The format() method is simply an object-oriented version of date(). It takes \$dateFormat as an argument. \$dateFormat is a string consisting of the same date formatting characters accepted by the procedural date() function.

4. Which of the following is the right way to use the DateTime class?

- a) `$date = get_Class(DateTime);`
- b) `$date = class DateTime;`
- c) `$date = new DateTime();`
- d) `$date = new class DateTime();`

Answer: c

Explanation: The way you use the DateTime class is like any other class: instantiate an object, and store it in a variable.

5. What will be the output of the following PHP code if date is 24/02/2008?

```
1.      <?php
2.      $date = new DateTime();
3.      echo $date->format('l,F,j,s,Y');
4.      ?>
```

- a) Sunday, February 24th 2008
- b) Sunday, 02 24 2008
- c) Sunday, 24 02 2008
- d) Sunday, 24th February 2008

Answer: a

Explanation: The `format()` method displays the date in same way as standard date function().

6. Which of the following statements can be used to set the time zone in individual scripts?

- a) `date_set_timezone('Europe/London');`
- b) `date_default_timezone_set('Europe/London');`
- c) `date_set_default_timezone('Europe/London');`
- d) `date_default_timezone('Europe/London');`

Answer: b

Explanation: The function `date_default_timezone_set` is used to set the default time zone used by all date/time functions in a script. You can also use `ini_set('date.timezone', 'Europe/London');`

- a) Only i)
- b) Only ii)
- c) i) and iv)
- d) iii) and iv)

Answer: c

Explanation: `listAbbreviations()` and `listIdentifiers()` are static methods.

- a) Only i)
- b) Only ii)
- c) i), ii), iii) and iv)
- d) iii) and iv)

Answer: c

Explanation: All of the given methods are non static.

9. Which of the following statements can be used to add two months to the existing date?

- a) `$date->modify('+2 months');`
- b) `$date = modify('+2 months');`
- c) `$date = modify('2+ months');`
- d) `$date->modify('2+ months');`

Answer: a

Explanation: To change the date stored by a `DateTime` object after it has been created, you use `DateTime::modify()` with a natural language expression.

10. Which method enables you to calculate whether daylight saving time is in force at a specific date and time?

- a) `getOffset()`
- b) `getTranitions()`
- c) `ISODate()`
- d) `savingTime()`

Answer: b
Explanation: This outputs a multidimensional array listing past and future changes to the offset from UTC for a DateTimeZone object.

- 1. Which two predefined variables are used to retrieve information from forms?**
- a) \$GET & \$SET**
 - b) \$_GET & \$_SET**
 - c) \$__GET & \$__SET**
 - d) GET & SET**

Answer: b
Explanation: The global variables \$_GET is used to collect form data after submitting an HTML form with the method="get". The variable \$_SET is also used to retrieve information from forms.

- 2. The attack which involves the insertion of malicious code into a page frequented by other users is known as _____**
- a) basic sql injection**
 - b) advanced sql injection**
 - c) cross-site scripting**
 - d) scripting**

Answer: c
Explanation: The cross-site scripting attack is among one of the top five security attacks carried out across the Internet. It is also known as XSS, this attack is a type of code injection attack which is made possible by incorrectly validating user data, which usually gets inserted into the page through a web form or using an altered hyperlink.

- 3. When you use the \$_GET variable to collect data, the data is visible to _____**
- a) none**
 - b) only you**
 - c) everyone**
 - d) selected few**

Answer: c
Explanation: The information sent from a form with the method GET is visible to everyone i.e. all variable names and values are displayed in the URL.

- 4. When you use the \$_POST variable to collect data, the data is visible to _____**
- a) none**
 - b) only you**
 - c) everyone**
 - d) selected few**

Answer: b
Explanation: The information sent from a form with the method POST is invisible to others i.e. all names/values are embedded within the body of the HTTP request.

- 5. Which variable is used to collect form data sent with both the GET and POST methods?**
- a) \$BOTH**
 - b) \$_BOTH**
 - c) \$REQUEST**
 - d) \$_REQUEST**

Answer: d
Explanation: In PHP the global variable \$_REQUEST is used to collect data after submitting an HTML form.

- 6. Which one of the following should not be used while sending passwords or other sensitive information?**
- a) GET**
 - b) POST**

- c) REQUEST
- d) NEXT

Answer: a

Explanation: The information sent from a form with the method GET is visible to everyone i.e. all variable names and values are displayed in the URL. So, it should not be used while sending passwords or other sensitive information.

7. Which function is used to remove all HTML tags from a string passed to a form?

- a) remove_tags()
- b) strip_tags()
- c) tags_strip()
- d) tags_remove()

Answer: b

Explanation: The function strip_tags() is used to strip a string from HTML, XML, and PHP tags.

8. What will be the value of the variable \$input in the following PHP code?

```
1.      <?php
2.      $input = "Swapna<td>Lawrence</td>you are really<i>pretty</i>!";
3.      $input = strip_tags($input, "<i></i>");
4.      echo $input;
5.      ?>
```

- a) Swapna Lawrence you are really pretty!
- b) Swapna <td>Lawrence</td> you are really<i>pretty</i>!
- c) Swapna <td>Lawrence</td> you are really pretty!
- d) Swapna Lawrence you are really<i>pretty</i>!

Answer: d

Explanation: Italic tags <i></i> might be allowable, but table tags <td></td> could potentially wreak havoc on a page.

9. To validate an email address, which flag is to be passed to the function filter_var()?

- a) FILTER_VALIDATE_EMAIL
- b) FILTER_VALIDATE_MAIL
- c) VALIDATE_EMAIL
- d) VALIDATE_MAIL

Answer: a

Explanation: The FILTER_VALIDATE_EMAIL is used to validate an e-mail address.

10. How many validation filters like FILTER_VALIDATE_EMAIL are currently available?

- a) 5
- b) 6
- c) 7
- d) 8

Answer: c

Explanation: There are seven validation filters. They are FILTER_VALIDATE_EMAIL, FILTER_VALIDATE_BOOLEAN, FILTER_VALIDATE_FLOAT, FILTER_VALIDATE_INT, FILTER_VALIDATE_IP, FILTER_VALIDATE_REGEXP, FILTER_VALIDATE_URL.

1. How many predefined variables does PHP use to authenticate a user?

- a) 1

- b) 2
- c) 3
- d) 4

Answer: b

Explanation: The variables PHP use to authenticate a user are `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']`.

2. Which of the following variables does PHP use to authenticate a user?

```
i) $_SERVER['PHP_AUTH_USER'] .
ii) $_SERVER['PHP_AUTH_USERS'] .
iii) $_SERVER['PHP_AUTH_PU'] .
iv) $_SERVER['PHP_AUTH_PW'] .
```

- a) i) and ii)
- b) ii) and iv)
- c) i) and iv)
- d) ii) and iii)

Answer: c

Explanation: `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']` store the username and password values, respectively.

- a) i) and iv)
- b) ii) and iv)
- c) ii) and iii)
- d) i) and iii)

Answer: a

Explanation: The function `isset()` is used to check whether a variable is set or not and the function `header()` sends a raw HTTP header to a client.

4. Which function is used to verify whether a variable contains a value?

- a) `header()`
- b) `footer()`
- c) `inset()`
- d) `isset()`

Answer: d

Explanation: The `isset()` function determines whether a variable has been assigned a value. Its prototype follows: `boolean isset(mixed var [,mixed var [,...]])`.

- a) ii) and iii)
- b) i) and iv)
- c) i), ii), iii) and iv)
- d) Only iv)

Answer: c

Explanation: The method PEAR'S HTTP authentication is used to provides a framework for user authentication on the HTTP. The data-based authentication is the process of confirming that a user who is attempting to log in to a database is authorized to do so. In the file-based authentication as some small sites does not have a need for database back-end to store data, but security is still important either the site is big or small. They need to authenticate some folder or file and want to set access credentials for that. One can handle such by using file-based authentication using PHP. The simplest way to restrict resource access is by hard-coding the username and password directly into the script.

6. In which authentication method does changing the username or password can be done only by entering the code and making the manual adjustment.

- a) Hard-coding a login pair directly into the script

- b) File-based authentication
- c) Data-based authentication
- d) PEAR'S HTTP authentication

Answer: a

Explanation: The simplest way to restrict resource access is by hard-coding the username and password directly into the script. In this authentication method, changing the username or password can be done only by entering the code and making the manual adjustment. This is one of the drawbacks of hard-coding a login pair directly into the script.

7. The authenticationFile.txt, the file which stores username and password should be stored ____ the server document root.

- a) Inside
- b) Outside
- c) Within
- d) None of the mentioned

Answer: b

Explanation: If it is not stored outside, the attacker could discover the file through brute force guessing.

8. Which function is used to split a string into a series of substrings, with each string boundary is determined by a specific separator?

- a) break()
- b) divide()
- c) explode()
- d) md5()

Answer: c

Explanation: Although they are a similar function, you should use explode() instead of split(). In fact split() function has been deprecated altogether.

- a) i) and iv)
- b) i) and iii)
- c) ii) and iii)
- d) ii) and iv)

Answer: c

Explanation: Such requirements are better satisfied by implementing a database based solution.

10. Which is the most powerful authentication method among the four?

- a) Hard-coding a login pair directly into the script
- b) File-based authentication
- c) Data-based authentication
- d) PEAR'S HTTP authentication

Answer: c

Explanation: It not only enhances administrative convenience and scalability but also can be integrated into a larger database infrastructure.

1. Which directive determines whether PHP scripts on the server can accept file uploads?

- a) file_uploads
- b) file_upload
- c) file_input
- d) file_intake

Answer: a

Explanation: With PHP, it is easy to upload files to the server. We need to ensure that PHP is configured to allow file uploads. In the "php.ini" file, search for the file_uploads directive, and set it to On. By default, its value is on.

2. Which of the following directive determines the maximum amount of time that a PHP script will spend attempting to parse input before registering a fatal error?

- a) max_take_time
- b) max_intake_time
- c) max_input_time
- d) max_parse_time

Answer: c

Explanation: This is relevant because particularly large files can take some time to upload, eclipsing the time set by this directive.

3. What is the default value of max_input_time directive?

- a) 30 seconds
- b) 60 seconds
- c) 120 seconds
- d) 1 second

Answer: b

Explanation: The default value of the max_input_time directive is 60 seconds.

4. Since which version of PHP was the directive max_file_limit available.

- a) PHP 5.2.1
- b) PHP 5.2.2
- c) PHP 5.2.12
- d) PHP 5.2.21

Answer: c

Explanation: The max_file_limit directive sets an upper limit on the number of files which can be simultaneously uploaded.

5. What is the default value of the directive max_file_limit?

- a) 10 files
- b) 15 files
- c) 20 files
- d) 25 files

Answer: c

Explanation: The default value of the directive max_file_limit is 20 files.

6. Which directive sets a maximum allowable amount of memory in megabytes that a script can allow?

- a) max_size
- b) post_max_size
- c) max_memory_limit
- d) memory_limit

Answer: d

Explanation: Its default value is 16M.

7. If you want to temporarily store uploaded files in the /tmp/phpuploads/ directory, which one of the following statement will you use?

- a) upload_tmp_dir "/tmp/phpuploads/ directory"
- b) upload_dir "/tmp/phpuploads/ directory"
- c) upload_temp_dir "/tmp/phpuploads/ directory"
- d) upload_temp_director "/tmp/phpuploads/ directory"

Answer: a

Explanation: Anyone can temporarily store uploaded files on the given directory. One cannot change upload_tmp_dir at the runtime. By the time a script runs, the upload process has already occurred.

8. Which superglobal stores a variety of information pertinent to a file uploaded to the server via a PHP script?

- a) `$_FILE` Array
- b) `$_FILES` Array
- c) `$_FILES_UPLOADED` Array
- d) `$_FILE_UPLOADED` Array

Answer: b

Explanation: The superglobal `$_FILES` is a two-dimensional associative global array of items which are being uploaded by via HTTP POST method and holds the attributes of files.

9. How many items are available in the `$_FILES` array?

- a) 2
- b) 3
- c) 4
- d) 5

Answer: d

Explanation: `$_FILES['userfile']['error']`, `$_FILES['userfile']['name']`, `$_FILES['userfile']['size']`, `$_FILES['userfile']['tmp_name']`, `$_FILES['userfile']['type']` are the five items in the array.

10. Which function is used to determine whether a file was uploaded?

- a) `is_file_uploaded()`
- b) `is_uploaded_file()`
- c) `file_uploaded("filename")`
- d) `uploaded_file("filename")`

Answer: b

Explanation: The function `is_uploaded_file()` checks whether the specified file is uploaded via HTTP POST. The syntax is `is_uploaded_file(file)`.

1. What is the full form of DNS?

- a) Digital Network System
- b) Domain Network System
- c) Digital Name System
- d) Domain Name System

Answer: d

Explanation: DNS stands for domain name system. It is the way that internet domain names are located and translated into internet protocol (IP) addresses. For example, if someone types `abc.com` into the web browser, a server behind the scenes will map that name to the IP address `206.16.49.139`.

2. Which one of the following function checks for the existence of DNS records?

- a) `checkdns()`
- b) `checkdnsr()`
- c) `checkdnsrr()`
- d) `checkdnsa()`

Answer: c

Explanation: The function `checkdnsrr()` is used to check DNS records for type corresponding to host. DNS records are checked based on the supplied host value and optional DNS resource record type, returning `TRUE` if any records are located and `FALSE` otherwise.

3. Which one of the following function is used to return an array consisting of various DNS resource records pertinent to a specific domain?

- a) `dns_get_record()`
- b) `dns_record()`
- c) `dnsrr_get_record()`
- d) `dnsrr_record()`

Answer: a

Explanation: The function `dns_get_record()` is used to get the DNS resource records associated with the specified hostname. This function is used to return an array consisting of various DNS resource records pertinent to a specific domain.

4. Which one of the following function is used to retrieve the MX records for the domain specified by hostname?

- a) `getmx()`
- b) `retrieve_mx()`
- c) `getmxrr()`
- d) `retrieve_mxrr()`

Answer: c

Explanation: The function `getmxrr()` returns the MX records for the specified internet host name. This function is used to retrieve the MX records for the domain specified by hostname.

5. What is the default port number of HTTPs?

- a) 70
- b) 80
- c) 90
- d) 100

Answer: b

Explanation: By default, The port number HTTP uses is port 80 and HTTPS uses port 443, but a URL like `http://www.abc.com:8080/path/` specifies that the web browser connects instead to port 8080 of the HTTP servers.

6. Which one of the following function returns the port number of a specified service?

- a) `getportname()`
- b) `getservername()`
- c) `getserverbyname()`
- d) `getservbyname()`

Answer: d

Explanation: The function `getservbyname()` returns the port number for a given Internet service and protocol. Example- `getservbyname("http", "tcp")` will return 80.

7. Which one of the following statements can be used to establish port 80 connection with `www.nachi.com`?

- a) `fsockopen("www.nachi.com", 80);`
- b) `sockopen(80,"www.nachi.com");`
- c) `fsockopen(80,"www.nachi.com");`
- d) `sockopen("www.nachi.com", 80);`

Answer: a

Explanation: The `fsockopen()` function establishes a connection to the resource designated by target on port.

8. Which one of the following function is used to send an email using PHP script?

- a) `mail_send()`
- b) `send_mail()`
- c) `mailrr()`
- d) `mail()`

Answer: d

Explanation: The function `mail()` allows you to send emails directly from a script. Using `mail(string to, string subject, string message)`, you can send any mail.

9. How many configuration directives pertinent to PHP's mail function are available?

- a) 4
- b) 5
- c) 6

d) 7

Answer: b

Explanation: They are- SMTP, sendmail_from, sendmail_path, smtp_port, mail.force_extra_parameters.

10. Which of the following statements is used to add an attachment to the mail?

- a) \$mime mail->attachment('attachment.pdf');
- b) \$mime mail=>attachment('attachment.pdf');
- c) \$mime mail->addAttachment('attachment.pdf');
- d) \$mime mail=>addAttachment('attachment.pdf');

Answer: c

Explanation: Call the Mail_Mime object's addAttachment() method passing in the attachment name and extension.

1. Which one of the following is the very first task executed by a session enabled page?

- a) Delete the previous session
- b) Start a new session
- c) Check whether a valid session exists
- d) Handle the session

Answer: c

Explanation: The session variables are set with the PHP global variable which is \$_SESSION. The very first task executed by a session enabled page is Check whether a valid session exists.

2. How many ways can a session data be stored?

- a) 3
- b) 4
- c) 5
- d) 6

Answer: b

Explanation: Within flat files(files), within volatile memory(mm), using the SQLite database(sqlite), or through user defined functions(user).

3. Which directive determines how the session information will be stored?

- a) save_data
- b) session.save
- c) session.save_data
- d) session.save_handler

Answer: d

Explanation: The class SessionHandler is used to wrap whatever internal save handler is set as defined by the session.save_handler configuration directive which is usually files by default.

4. Which one of the following is the default PHP session name?

- a) PHPSESSID
- b) PHPSESID
- c) PHPSESSIONID
- d) PHPIDSESS

Answer: a

Explanation: You can change this name by using the session.name directive.

5. If session.use_cookie is set to 0, this results in use of _____

- a) Session
- b) Cookie
- c) URL rewriting
- d) Nothing happens

Answer: c

Explanation: The URL rewriting allows to completely separate the URL from the resource. URL rewriting can turn unsightly URLs into nice ones with a lot less agony and expense than picking a good domain name. It enables to fill out your URLs with friendly, readable keywords without affecting the underlying structure of pages.

6. If the directive `session.cookie_lifetime` is set to 3600, the cookie will live until _____

- a) 3600 sec**
- b) 3600 min**
- c) 3600 hrs**
- d) the browser is restarted**

Answer: a

Explanation: The lifetime is specified in seconds, so if the cookie should live 1 hour, this directive should be set to 3600.

7. Neglecting to set which of the following cookie will result in the cookie's domain being set to the host name of the server which generated it.

- a) `session.domain`**
- b) `session.path`**
- c) `session.cookie_path`**
- d) `session.cookie_domain`**

Answer: d

Explanation: The directive `session.cookie_domain` determines the domain for which the cookie is valid.

8. What is the default number of seconds that cached session pages are made available before the new pages are created?

- a) 360**
- b) 180**
- c) 3600**
- d) 1800**

Answer: b

Explanation: The directive which determines this is `session.cache_expire`.

9. What is the default time(in seconds) for which session data is considered valid?

- a) 1800**
- b) 3600**
- c) 1440**
- d) 1540**

Answer: c

Explanation: The `session.gc_maxlifetime` directive determines this duration. It can be set to any required value.

10. Which one of the following function is used to start a session?

- a) `start_session()`**
- b) `session_start()`**
- c) `session_begin()`**
- d) `begin_session()`**

Answer: b

Explanation: A session is started with the function `session_start()`. The session variables are set with the PHP global variable which is `$_SESSION`.

1. Which function is used to erase all session variables stored in the current session?

- a) `session_destroy()`**
- b) `session_change()`**
- c) `session_remove()`**
- d) `session_unset()`**

Answer: d

Explanation: The function `session_unset()` frees all session variables that is currently registered. This will not completely remove the session from the storage mechanism. If you want to completely destroy the session, you need to use the function `session_destroy()`.

2. What will the function `session_id()` return if no parameter is passed?

- a) Current Session Identification Number
- b) Previous Session Identification Number
- c) Last Session Identification Number
- d) Error

Answer: a

Explanation: The function `session_id()` will return the session id for the current session or the empty string ("") if there is no current session.

3. Which one of the following statements should you use to set the session username to Nachi?

- a) `$SESSION['username'] = "Nachi";`
- b) `$_SESSION['username'] = "Nachi";`
- c) `session_start("nachi");`
- d) `$SESSION_START["username"] = "Nachi";`

Answer: b

Explanation: You need to refer the session variable 'username' in the context of the `$_SESSION` superglobal.

4. What will be the output of the following PHP code? (Say your previous session username was nachi.)

```
1.      unset($_SESSION['username']);  
2.      printf("Username now set to: %s", $_SESSION['username']);
```

- a) Username now set to: nachi
- b) Username now set to: System
- c) Username now set to:
- d) Error

Answer: c

Explanation: If someone want to destroy a single session variable then they can use the function `unset ()` to unset a session variable. To delete the session variable 'username' we use the `unset ()` function.

5. An attacker somehow obtains an unsuspecting user's SID and then using it to impersonate the user in order to gain potentially sensitive information. This attack is known as _____

- a) session-fixation
- b) session-fixing
- c) session-hijack
- d) session-copy

Answer: a

Explanation: The attack session fixation attempts to exploit the vulnerability of a system that allows one person to set another person's session identifier. You can minimize this risk by regenerating the session ID on each request while maintaining the session-specific data. PHP offers a convenient function named `session_regenerate_id()` that will replace the existing ID with a new one.

6. Which parameter determines whether the old session file will also be deleted when the session ID is regenerated?

- a) `delete_old_file`
- b) `delete_old_session`
- c) `delete_old_session_file`
- d) `delete_session_file`

Answer: b

Explanation: The parameter `delete_old_session` determines whether the old session file will also be deleted when the session ID is regenerated.

7. Which function effectively deletes all sessions that have expired?

- a) `session_delete()`
- b) `session_destroy()`
- c) `session_garbage_collect()`
- d) `SessionHandler::gc`

Answer: d

Explanation: `SessionHandler::gc` is used to clean up expired sessions. It is called randomly by PHP internally when a `session_start()` is invoked.

8. Which function is used to transform PHP's session-handler behavior into that defined by your custom handler?

- a) `session_set_save()`
- b) `session_set_save_handler()`
- c) `Session_handler()`
- d) `session_save_handler()`

Answer: b

Explanation: The function `session_set_save_handler()` is used to set the user-level session storage functions which are used for storing and retrieving data associated with a session.

9. The `session_start()` function must appear _____

- a) after the `html` tag
- b) after the `body` tag
- c) before the `body` tag
- d) before the `html` tag

Answer: d

Explanation: Like this: `<?php session_start(); ?> <html>`

10. What is the return type of `session_set_save_handler()` function?

- a) `boolean`
- b) `integer`
- c) `float`
- d) `character`

Answer: a

Explanation: Returns `TRUE` on success or `FALSE` on failure.

1. Which one of the following statements should be used to disable just the `fopen()`, and `file()` functions?

- a) `disable_functions = fopen(), file()`
- b) `disable_functions = fopen, file`
- c) `functions_disable = fopen(), file()`
- d) `functions_disable = fopen, file`

Answer: b

Explanation: You can set `disable_functions` equal to a comma-delimited list of function names that you want to disable.

2. Which one of the following statements should be used to disable the use of two classes `administrator` and `janitor`?

- a) `disable_classes = "administrator, janitor"`
- b) `disable_classes = class administrator, class janitor`
- c) `disable_classes = class "administrator", class "janitor"`
- d) `disable_class = class "administrator", class "janitor"`

Answer: a

Explanation: There may be classes inside the libraries that you'd rather not make available. You can prevent the use of these classes with the `disable_classes` directive.

3. What is the default value of `max_execution_time` directive? This directive specifies how many seconds a script can execute before being terminated.

- a) 10
- b) 20
- c) 30
- d) 40

Answer: c

Explanation: This can be useful to prevent users' scripts from consuming too much CPU time. If `max_execution_time` is set to 0, no time limit will be set.

4. The `memory_limit` is only applicable if _____ is enabled when you configure PHP.

- a) `-enable-limit`
- b) `-enable-memory-limit`
- c) `-enable-memory-limit`
- d) `-memory-limit`

Answer: c

Explanation: This directive specifies, in megabytes, how much memory a script can use. Default value: 128M.

5. Suppose all web material is located within the directory `/home/www`. To prevent users from viewing and manipulating files such as `/etc/passwd`, which one of the following statements should you use?

- a) `open_dir = "/home/www"`
- b) `open_dir = /home/www/`
- c) `open_basedir = /home/www/`
- d) `open_basedir = "/home/www"`

Answer: d

Explanation: We can use `open_basedir` statement to define the paths from which we can access files using functions like `fopen()` and `gzopen()`. It is used to prevent users from viewing and manipulating files such as `/etc/passwd`.

6. Which Apache directive outputs Apache's server version, server name, port and compile-in modules?

- a) `ServerSignature`
- b) `ServerName`
- c) `ServerDetails`
- d) `ServerInfo`

Answer: a

Explanation: It is capable of displaying output like: Apache/2.2.11 (Ubuntu) Server at localhost Port 80.

7. Which directive determines which degree of server details is provided if the `ServerSignature` directive is enabled?

- a) `ServerAddons`
- b) `ServerExtra`
- c) `ServerTokens`
- d) `ServerDetails`

Answer: c

Explanation: Six options are available: Full, Major, Minimal, Minor, OS, and Prod. If it is set to Full 'Apache/2.2.11 (Ubuntu) PHP/5.3.2 Server' will be displayed.

8. Which directive should we disable to obscure the fact that PHP is being used on our server?

- a) `show_php`
- b) `expose_php`
- c) `print_php`
- d) `info_php`

Answer: b

Explanation: Using the `expose_php` directive we can prevent PHP version details from being appended on our web server signature. When `expose_php` is disabled, the server signature will look like: `Apache/2.2.11 (Ubuntu) Server`.

9. Say I want to change the extension of a PHP file, which of the following statements should I edit to change from .php to .html in the `httpd.conf` file?

- a) `AddType application/x-httpd-php .php`
- b) `AddType application/x-httpd-php .asp`
- c) `AddType application/x-httpd-asp .php`
- d) `AddType application/x-httpd-asp .asp`

Answer: a

Explanation: Just change the .php to .html and your extension will change. This is used for security purpose.

10. The developers of PHP deprecated the safe mode feature as of which PHP version.

- a) PHP 5.1.0
- b) PHP 5.2.0
- c) PHP 5.3.0
- d) PHP 5.3.1

Answer: c

Explanation: This happened because safe mode often creates many problems as it resolves, largely due to the need for enterprise applications to use many of the features safe mode disables.

1. What will be the output of the following PHP code?

```
1.      <?php
2.      $number = array(0,1,two,three,four,5);
3.      $num = preg_grep("/[0-5]/", $number);
4.      print_r($num);
5.      ?>
```

- a) `Array([0]=>0 [1]=>1 [2]=>two [3]=>three [4]=>four [5]=>5)`
- b) `Array([2]=>two [3]=>three [4]=>four)`
- c) `Array([1]=> 1)`
- d) `Array([0]=>0 [1]=>1 [5]=>5)`

Answer: d

Explanation: The `preg_grep` function is used to search an array for specific patterns and then return a new array based on that filtering.

2. What will be the output if we replace the line `$num = preg_grep("/[0-5]/", $number);` with `$num = preg_grep("/[0-5]/", $number, PREG_GREP_INVERT);`?

- a) `Array([0]=>0 [1]=>1 [2]=>two [3]=>three [4]=>four [5]=>5)`
- b) `Array([2]=>two [3]=>three [4]=>four)`
- c) `Array([1]=> 1)`
- d) `Array([0]=>0 [5]=>5)`

Answer: b

Explanation: When we include `PREG_GREP_INVERT`, this will invert our data, so instead of outputting numbers it will output our non-numeric values.

3. Which one of the following functions are used to search a string?

- a) `preg_match`
- b) `preg_search`

- c) preg_find
- d) preg_found

Answer: a

Explanation: The function preg_match() searches string for pattern and it returns true if pattern exists, and false otherwise. The function returns 1 if search was successful else returns 0.

4. What will be the output of the following PHP code?

```
1.      <?php
2.      $name = "What is your name?";
3.      if (preg_match("/name/"),$name)
4.      echo "My name is Will Pitt ";
5.      else
6.      echo "My name is not Will Pitt ";
7.      if (preg_match("/are/"))
8.      echo "I am great";
9.      else
10.     echo "I am not great";
11.     ?>
```

- a) My name is Will Pitt I am great
- b) My name is not Will Pitt I am great
- c) My name is Will Pitt I am not great
- d) My name is not Will Pitt I am not great

Answer: c

Explanation: The code uses preg_match to check for a keyword and replies based on whether it is true (1) or false (0).

5. Which one of the following preg PHP function is used to do a find and replace on a string or an array?

- a) preg_replace()
- b) preg_find()
- c) preg_find_replace()
- d) preg_findre()

Answer: a

Explanation: In preg_replace() function, after the replacement has occurred, the modified string will be returned and if no matches are found, the string will remain unchanged.

6. What will be the output of the following PHP code?

```
1.      <?php
2.      $str = "Hello! My name is Cameron Fox. Coffee?";
3.      $find = array('/is/', '/coffee/');
4.      $replace = array('/was/', '/tea/');
5.      echo preg_replace ($find, $replace, $str);
6.      ?>
```

- a) Hello! My name was Cameron Fox. tea?
- b) Hello! My name is Cameron Fox. tea?
- c) Hello! My name is Cameron Fox. Coffee?
- d) Hello! My name was Cameron Fox. Coffee?

Answer: d

Explanation: Coffee was not replaced because the preg_replace function is case sensitive. Therefore it treats coffee and Coffee differently.

7. Which one of the following preg PHP functions is used to take a string, and put it in an array?

- a) preg_destroy()
- b) preg_split()
- c) preg_unchain()
- d) preg_divide()

Answer: b

Explanation: The string is broken up into different values in the array based upon your input.

8. What will be the output of the following PHP code?

```
1.      <?php
2.      $line = "You like dogs. I hate dogs. We should marry.";
3.      $sen = preg_split('/\./', $line);
4.      print_r($sen);
5.      ?>
```

- a) You like dogs. I hate dogs. We should marry.
- b) Array([0]=>You like dogs. I hate dogs. We should marry.)
- c) Array([0]=>You like dogs. [1]=>I hate dogs. [2]=>We should marry.)
- d) Error

Answer: c

Explanation: We use a '.' period to split the data, therefor giving each sentence it's own array entry.

9. Which one of the following is not a preg PHP function?

- a) preg_match
- b) preg_match_all
- c) preg_matchall
- d) preg_split

Answer: c

Explanation: The function preg_match_all() matches all occurrences of pattern in string.

10. Parameter flags was added in which version of PHP?

- a) PHP 4.0
- b) PHP 4.1
- c) PHP 4.2
- d) PHP 4.3

Answer: d

Explanation: Parameter flags was added in PHP 4.3 version.

1. Which one of the following databases has PHP supported almost since the beginning?

- a) Oracle Database
- b) SQL

- c) SQL+
- d) MySQL

Answer: d

Explanation: We can connect, insert, update, delete and retrieve data from the databases with the help of PHP and MySQL is the most popular database system used with PHP.

2. The updated MySQL extension released with PHP 5 is typically referred to as _____

- a) MySQL
- b) mysql
- c) mysqli
- d) mysqlly

Answer: c

Explanation: The updated MySQL extension with PHP 5 is known as MySQL and typically referred to as mysqli.

3. Which one of the following lines need to be uncommented or added in the php.ini file so as to enable mysqli extension?

- a) extension=php_mysqli.dll
- b) extension=mysql.dll
- c) extension=php_mysqli.dl
- d) extension=mysqli.dl

Answer: a

Explanation: Also make sure that extension_dir directive points to the appropriate directory.

4. In which version of PHP was MySQL Native Driver(also known as mysqlnd) introduced?

- a) PHP 5.0
- b) PHP 5.1
- c) PHP 5.2
- d) PHP 5.3

Answer: d

Explanation: PHP required that MySQL client library be installed on the server from which PHP was communicating with MySQL, whether the MySQL server also happened to reside locally or elsewhere. PHP 5.3 removes this problem by introducing MySQL Native Driver.

5. Which one of the following statements is used to create a table?

- a) CREATE TABLE table_name (column_name column_type);
- b) CREATE table_name (column_type column_name);
- c) CREATE table_name (column_name column_type);
- d) CREATE TABLE table_name (column_type column_name);

Answer: a

Explanation: The example creates a table called "student" that contains five columns: SID, LastName, FirstName, Address, and City:

```
CREATE TABLE student (  
  
    SID int,  
  
    LastName varchar(255),  
  
    FirstName varchar(255),  
  
    Address varchar(255),  
  
    City varchar(255)  
);
```

6. Which one of the following statements instantiates the mysqli class?

- a) `mysqli = new mysqli()`
- b) `$mysqli = new mysqli()`
- c) `$mysqli->new(mysqli)`
- d) `mysqli->new(mysqli)`

Answer: b

Explanation: If you choose to interact with MySQL server using the object-oriented interface, you need to first instantiate the mysqli class via its constructor.

7. Which one of the following statements can be used to select the database?

- a) `$mysqli=select_db('databasename');`
- b) `mysqli=select_db('databasename');`
- c) `mysqli->select_db('databasename');`
- d) `$mysqli->select_db('databasename');`

Answer: d

Explanation: `$mysqli->select_db('databasename');` can be used to select the database.

8. Which one of the following methods can be used to diagnose and display information about a MySQL connection error?

- a) `connect_errno()`
- b) `connect_error()`
- c) `mysqli_connect_errno()`
- d) `mysqli_connect_error()`

Answer: c

Explanation: The mysqli extension includes a few features that can be used to capture error messages or alternatively you can use exceptions.

9. Which method returns the error code generated from the execution of the last MySQL function?

- a) `errno()`
- b) `errnumber()`
- c) `errorno()`
- d) `errornumber()`

Answer: a

Explanation: Error numbers are often used in lieu of natural-language message to ease software internationalization efforts and allow for customization of error messages.

10. If there is no error, then what will the error() method return?

- a) TRUE
- b) FALSE
- c) Empty String
- d) 0

Answer: c

Explanation: The function error is used to deal with error handling and logging. If there is no error, then the error() method will return an empty string.

1. Which one of the following statements should be used to include a file?

- a) `#include 'filename';`
- b) `include 'filename';`
- c) `@include 'filename';`
- d) `#include <filename>;`

Answer: b

Explanation: Include in PHP will takes all the code from the specified file and copies to the existing file containing

include statement. An example of this-

```
1. <?php
2. #include 'mysql.connect.php';
3. //begin database selection and queries.
4. ?>
```

2. Which one of the following methods is responsible for sending the query to the database?

- a) query()
- b) send_query()
- c) sendquery()
- d) mysqli_query()

Answer: d

Explanation: The method mysqli_query() is responsible for sending the query to the database. Query() method was previously used in older versions of PHP.

3. Which one of the following methods recuperates any memory consumed by a result set?

- a) destroy()
- b) mysqli_free_result()
- c) alloc()
- d) free()

Answer: b

Explanation: The function mysqli_free_result() is used to free the memory which is associated with the result. Once this method is executed, the result set is no longer available. Free() function was used in the previous version of PHP.

4. Which of the methods are used to manage result sets using both associative and indexed arrays?

- a) get_array() and get_row()
- b) get_array() and get_column()
- c) fetch_array() and fetch_row()
- d) mysqli_fetch_array() and mysqli_fetch_row()

Answer: d

Explanation: The method mysqli_fetch_array() is used to fetch a result row as an associative array or a numeric array. And the function mysqli_fetch_row() is used to fetch one row from a result-set and returns it as an enumerated array. The method fetch_array() and fetch_row() were used in the previous version of PHP.

5. Which one of the following method is used to retrieve the number of rows affected by an INSERT, UPDATE, or DELETE query?

- a) num_rows()
- b) affected_rows()
- c) changed_rows()
- d) mysqli_affected_rows()

Answer: d

Explanation: The method mysqli_num_rows() is only useful for determining the number of rows retrieved by a SELECT query. But to retrieve the number of rows affected by INSERT, UPDATE, or DELETE query, use mysqli_affected_rows(). Num_rows() and affected_rows() were used in previous version of PHP.

6. Which version of MySQL introduced the prepared statements?

- a) MySQL 4.0
- b) MySQL 4.1
- c) MySQL 4.2
- d) MySQL 4.3

Answer: b

Explanation: When the query() method is looped repeatedly it comes at a cost of both overhead, because of the need to repeatedly parsing of the almost identical query for validity, and coding convenience, because of the need to repeatedly reconfigure the query using the new values for each iteration. To help resolve the issues incurred by repeatedly executed queries, MySQL introduced prepared statements.

7. Which of the following methods is used to execute the statement after the parameters have been bound?

- a) bind_param()**
- b) bind_result()**
- c) bound_param()**
- d) bound_result()**

Answer: a

Explanation: Once the statement has been prepared, it needs to be executed. Exactly when it's executed depends upon whether you want to work with bound parameters or bound results. In the case of bound parameters, you'd execute the statement after the parameters have been bound with the bind_param() method.

8. Which one of the following methods is used to recuperating prepared statements resources?

- a) end()**
- b) finish()**
- c) mysqli_close()**
- d) close()**

Answer: c

Explanation: The function mysqli_close() is used to close an opened database connection. Once you've finished using a prepared statement, the resources it requires can be recuperated with the mysqli_close() method. Close() was used in previous version of PHP.

9. Which method retrieves each row from the prepared statement result and assigns the fields to the bound results?

- a) get_row()**
- b) fetch_row()**
- c) fetch()**
- d) mysqli_fetch_row()**

Answer: d

Explanation: The function mysqli_fetch_row() is used to fetch row from a result-set and returns it as an enumerated array.

Its syntax is mysqli_fetch_row(result);

10. Which method rolls back the present transaction?

- a) commit()**
- b) undo()**
- c) mysqli_rollback()**
- d) rollback()**

Answer: c

Explanation: The function mysqli_rollback() is used to roll back from the current transaction for the specified database connection. Its syntax is: mysqli_rollback(connection);

Rollback() was used in previous version of PHP.

1. Which one of the following is not a valid class name?

- a) ShopProduct**
- b) Shopproduct**
- c) Shopproduct1**
- d) 1shopproduct**

Answer: d

Explanation: You declare a class with the class keyword and an arbitrary class name. Class names can be any

combination of numbers and letters, although they must not begin with a number.

2. Fill in the blank with the best option. An Object is a/an _____ of a class.

- a) type
- b) prototype
- c) instance
- d) object

Answer: c

Explanation: An object is said to be an instance of its class. It is of the type defined by the class.

Answer: b

Explanation: In PHP object is created using new operator. \$product1 and \$product2 are different objects of the same type generated from a single class.

4. Which version of PHP introduced the visibility keywords i.e public, private, and protected?

- a) PHP 4
- b) PHP 5
- c) PHP 5.1
- d) PHP 5.3

Answer: b

Explanation: In PHP 4, all properties were declared with var keyword, which is identical in effect to using public. For the sake of backward compatibility, PHP 5 accepts var in place of public for properties.

5. Which characters is used to access property variables on an object-by-object basis?

- a) ::
- b) =
- c) ->
- d) .

Answer: c

Explanation: Example: \$product1->title="My Life";

6. Code that uses a class, function, or method is often described as the _____

- a) client code
- b) user code
- c) object code
- d) class code

Answer: a

Explanation: Code that uses a class, function, or method is often described as the class's, function, or method client or as client code.

7. Which keyword precedes a method name?

- a) method
- b) function
- c) public
- d) protected

Answer: b

Explanation: A method declaration resembles a function declaration. The function keyword precedes a method name, followed by an optional list of argument variables in parentheses.

8. If you omit the visibility keyword in your method declaration, by default the method will be declared as _____

- a) public
- b) private
- c) protected

d) friendly

Answer: a

Explanation: By declaring a method public, you ensure that it can be invoked from outside of the current object.

9. Which function is used to determine whether the variable's value is either TRUE or FALSE?

- a) boolean()**
- b) is_boolean()**
- c) bool()**
- d) is_bool()**

Answer: d

Explanation: None.

10. What will be the output of the following PHP code?

```
1. <?php
2.     class ShopProductWriter
3.     {
4.         public function write( $shopProduct )
5.         {
6.             $str = "{$shopProduct->title}: " . $shopProduct->getProducer() . " ({$shopProduct->getPrice()})";
7.             print $str;
8.         }
9.     }
10.     $product1 = new ShopProduct( "My Antonia", "Willa", "Cather", 5.99 );
11.     $writer = new ShopProductWriter();
12.     $writer->write( $product1 );
13. ?>
```

- a) Error**
- b) Cather: Willa My Antonia (5.99)**
- c) Willa: Cather My Antonia (5.99)**
- d) My Antonia: Willa Cather (5.99)**

Answer: d

Explanation: None.

1. Which version of PHP introduced class type hints?

- a) PHP 4**
- b) PHP 4.3**
- c) PHP 5**
- d) PHP 5.3**

Answer: c

Explanation: None.

2. Inheritance is the means by which one or more classes can be derived from a/an _____ class.

- a) base**
- b) abstract**

c) null

d) predefined

Answer: a

Explanation: A class that inherits from another is said to be a subclass of it. This relationship is often described in terms of parents and children. A child class is derived from and inherits characteristics from the parent.

3. What will be the output of the following PHP code?

```
1. <?php
2. class MyClass
3. {
4. }
5.
6. class NotMyClass
7. {
8. }
9. $a = new MyClass;
10.
11. var_dump($a instanceof MyClass);
12. var_dump($a instanceof NotMyClass);
13. ?>
```

4. What will be the output of the following PHP code?

```
1. <?php
2. class ParentClass
3. {
4. }
5.
6. class MyClass extends ParentClass
7. {
8. }
9.
10. $a = new MyClass;
11.
12. var_dump($a instanceof MyClass);
13. var_dump($a instanceof ParentClass);
14. ?>
```

5. What will be the output of the following PHP code?

```

1. <?php
2. class MyClass
3. {
4. }
5.
6. $a = new MyClass;
7. var_dump(!($a instanceof stdClass));
8. ?>

```

- a) bool(true)
- b) bool(false)
- c) error
- d) none of the mentioned

Answer: a

Explanation: To check if an object is not an instanceof a class, the logical not operator can be used.

6. What will be the output of the following PHP code?

```

1. <?php
2. interface MyInterface
3. {
4. }
5.
6. class MyClass implements MyInterface
7. {
8. }
9.
10. $a = new MyClass;
11.
12. var_dump($a instanceof MyClass);
13. var_dump($a instanceof MyInterface);
14. ?>

```

7. What should be used to refer to a method in the context of a class rather than an object you use?

- a) ->
- b) __
- c) \$
- d) ::

Answer: d

Explanation: Example- parent::__construct()

8. Prior to which version of PHP did constructors took the name of the enclosing class.

- a) PHP 4
- b) PHP 5
- c) PHP 5.3
- d) PHP 5.4

Answer: b

Explanation: The new unified constructors use the name `__construct()`. Using the old syntax, a call to a parent constructor would tie you to that particular class: `parent::ShopProduct()`;

9. Which method or property can only be accessed from within the enclosing class? Even subclasses have no access.

- a) public
- b) friendly
- c) private
- d) protected

Answer: c

Explanation: None.

10. A mutator method is also called as _____

- a) Setter
- b) Accessor
- c) Getter
- d) Destructor

Answer: a

Explanation: An accessor method is called getter. The common use of a mutator method is to initialise the value of member variables of a class.

1. When you are building administrative links you'll need to accept two arguments, which of the following are they?

- a) URL of previous entry and URL of the entry you are working with
- b) The current page and previous page
- c) URL of previous entry and previous page
- d) The current page and URL of the entry you are working with

Answer: d

Explanation: Your function should look like this:

```
1. function adminLinks($page, $url)
2. {
3.     //Build admin links here
4. }
```

2. Once your application can generate administrative links, you need to load those links into _____

- a) php.ini
- b) index.ini
- c) index.php
- d) start.php

Answer: c

Explanation: You place your administrative links only on the full display of an entry, so you must place the call to load information from `adminLinks()` within a conditional statement

3. The URLs in the administrative links won't mean anything to admin.php unless you modify _____

- a) .htaccess
- b) .adminaccess
- c) .htmlaccess

d) .htaccess

Answer: a

Explanation: You need to modify .htaccess with an additional rule that handles URLs passed in a link to admin.php.

4. The (|/) tells the server to match _____

- a) nothing**
- b) forward slash**
- c) backward slash**
- d) either nothing or a forward slash**

Answer: d

Explanation: The vertical pipe character (|) is the regular expression equivalent of “or”.

5. ([\w-]+) will match _____

- a) one word characters**
- b) one or more word characters**
- c) one or more word characters and/or hyphens**
- d) one or more word characters and hyphens**

Answer: c

Explanation: ([\w-]+), will match one or more word characters and/or hyphens—which is what your custom entry URLs consist.

6. You need to check whether _____ is set, to determine whether you’re editing an entry or creating a new one.

- a) \$_GET[‘url’]**
- b) \$_SET[‘url’]**
- c) \$_GET[‘admin’]**
- d) \$_SET[‘admin’]**

Answer: a

Explanation: If an entry is being edited, you need to load the existing entry data and save each piece in a variable.

7. To identify entries marked for deletion, you check whether \$_GET[‘page’] == ‘delete’ inside _____

- a) index.php**
- b) index.ini**
- c) admin.php**
- d) .htaccess**

Answer: c

Explanation: In admin.php, you check whether \$_GET[‘page’] == ‘delete’, then pass the entry URL to be deleted to a function.

8. To declare the function to confirm the deletion you need to add the code to _____

- a) inc.php**
- b) functions.inc.php**
- c) include.php**
- d) functions.include.php**

Answer: b

Explanation: You need to add the following code –

```
1. function confirmDelete($db, $url)
2. {
3.     $e = retrieveEntries($db, '', $url);
4.     return <<<FORM
```

```

5.      <form action="/simple_blog/admin.php" method="post">
6.      <fieldset>
7.      <legend>Are You Sure?</legend>
8.      <p>Are you sure you want to delete the entry "$e[title]"?</p>
9.      <input type="submit" name="submit" value="Yes" />
10.     <input type="submit" name="submit" value="No" />
11.     <input type="hidden" name="action" value="delete" />
12.     <input type="hidden" name="url" value="$url" />
13.     </fieldset>
14.     </form>
15.     FORM;
16. }

```

9. Your confirmation form submits your choice, via the _____ method, to _____

- a) GET index.php
- b) GET admin.php
- c) POST index.php
- d) POST admin.php

Answer: d

Explanation: To process this, you need to add an additional block of code to the top of admin.php that determines what choices you've made and act accordingly.

10. When a user confirms that he wishes to delete an entry, that entry's URL is passed to a function which removes the entry from the _____

- a) index.php
- b) function.inc.php
- c) database
- d) admin.php

Answer: c

Explanation: If the function is successful, you send the user to the main page. If it fails, you stop the execution of the script and display an error, letting the user know that something went wrong.

1. Before you can start processing images with PHP, you must first add the ability to upload images to your administrative form on _____

- a) .htaccess
- b) function.inc.php
- c) index.php
- d) admin.php

Answer: d

Explanation: To do this, you need to add a file upload input to your administrative form.

2. When you're uploading files you need to set the enctype of the form to _____

- a) text
- b) text/file
- c) multipart/form-data
- d) multimedia/form-data

Answer: c

Explanation: Set the enctype of the form to multipart/form-data, which can accept files and standard form values.

3. To check whether a file was uploaded, you look in the _____ superglobal array.

- a) \$_FILES**
- b) \$_DOCS**
- c) \$_DOCUMENTS**
- d) \$_FOLDERS**

Answer: a

Explanation: Whenever a file is uploaded via an HTML form, that file is stored in temporary memory and information about the file is passed in the \$_FILES superglobal.

4. To make the ImageHandler class portable you should create a separate file for it called _____

- a) imagehandler.inc.php**
- b) images.inc.php**
- c) handler.inc.php**
- d) imghandler.inc.php**

Answer: b

Explanation: You save this file in the inc folder (full path: /xampp/htdocs/simple_blog/inc/images.inc.php).

5. DocBlocks are indicated by opening a comment using _____

- a) /***
- b) /****
- c) /*****
- d) /*/**

Answer: c

Explanation: This is a special comment that provides information about a class, property, or method.

6. To process the file, you need to break the array from \$_FILES into individual values. You can do this using the _____ function.

- a) divide()**
- b) list()**
- c) break()**
- d) indi()**

Answer: b

Explanation: The list() function allows you to create named variables for each array index as a comma-separated list.

7. Before you try to process the file, you need to make sure that your \$err value is equivalent to _____

- a) UPLOAD_ERR_OK**
- b) UPLOAD_NO_ERR**
- c) UPLOAD_ERR_NO_OK**
- d) UPLOAD_ERR**

Answer: a

Explanation: When you're dealing with files uploaded through an HTML form, you have access to a special constant called UPLOAD_ERR_OK that tells you whether a file uploaded successfully.

8. You use the \$_SERVER superglobal and your _____ property to create your path to check.

- a) \$load_dir**
- b) \$load**
- c) \$save**
- d) \$save_dir**

Answer: d

Explanation: // Determines the path to check

```
$path = $_SERVER['DOCUMENT_ROOT'] . $this->save_dir;
```

9. Which function do you have to use to check whether the \$path you've stored exists?

- a) path_dir()
- b) path()
- c) is_dir()
- d) path_dir()

Answer: c

Explanation: If the path exists, is_dir() returns TRUE; otherwise, it returns FALSE.

10. Which one of the following is true about the following line – \$obj = new ImageHandler('/images/', array(400, 300));?

- a) This snippet sets the maximum dimensions allowed to 400 pixels wide by 300 pixels high
- b) This snippet sets the minimum dimensions allowed to 300 pixels wide by 400 pixels high
- c) This snippet sets the minimum dimensions allowed to 400 pixels wide by 300 pixels high
- d) This snippet sets the maximum dimensions allowed to 300 pixels wide by 400 pixels high

Answer: a

Explanation: If you needed to change the size of your images, you can change the dimensions using the above instantiation of ImageHandler.

1. Which version of PHP introduced the static keyword?

- a) PHP 4
- b) PHP 5
- c) PHP 5.2
- d) PHP 5.3

Answer: b

Explanation: The static keyword was introduced with PHP 5. It cannot be used in PHP 4 scripts.

2. Which keyword is used to access a static method or property from within the same class(rather than from child)?

- a) static
- b) strat
- c) self
- d) set

Answer: c

Explanation: Self is to classes what the \$this pseudo-variable is to objects.

3. In which of the following circumstance should you use a static reference to a non static method?

- a) Making a method call using parent
- b) Making a method call using child
- c) Making an object call using parent
- d) Making an object call using child

Answer: a

Explanation: Making a method call using parent is the only circumstance in which you should use a static reference to a non-static method.

4. Which one of the following variable cannot be used inside a static method?

- a) \$this
- b) \$get
- c) \$set
- d) \$date

Answer: a

Explanation: By definition, static methods are not invoked in the context of an object. For this reason, static methods and properties are often referred to as class variables and properties.

5. What does PDO stand for?

- a) PHP Data Orientation**
- b) PHP Database Object**
- c) PHP Database Orientation**
- d) PHP Data Object**

Answer: d

Explanation: The PDO class provides a common interface to different database applications.

6. Which version of PHP allows you to define constant properties within a class?

- a) PHP 4**
- b) PHP 4.1**
- c) PHP 4.3**
- d) PHP 5**

Answer: d

Explanation: None.

7. Which keyword is used to declare a constant property?

- a) const**
- b) con**
- c) constant**
- d) _constant**

Answer: a

Explanation: A constant property is declared with the const keyword. Like global constants, class constants cannot be changed once they are set.

8. Which one of the following is a constant variable?

- a) const \$name**
- b) const \$NAME**
- c) constant NAME**
- d) const NAME**

Answer: d

Explanation: Constants are not prefixed with a dollar sign like regular properties. By convention, they are often named using only uppercase characters

9. What will happen if you try to set a value to a constant once it has been declared?

- a) The value of the variable will change**
- b) The value of the variable will not change**
- c) Parse Error**
- d) Nothing**

Answer: c

Explanation: You should use constants when your property needs to be available across all instances of a class, and when the property value needs to be fixed and unchanging.

- a) 1**
- b) 2**
- c) 3**
- d) 4**

Answer: d

Explanation: All scalar data i.e boolean, integer, float and string can be contained in constants.

1. Which one of the following class can not be instantiated?

- a) inherited class**
- b) abstract class**
- c) constant class**
- d) every class**

Answer: b

Explanation: An abstract class cannot be instantiated. Instead, it defines (and, optionally, partially implements) the interface for any class that might extend it.

2. Which one of the following keywords are used to define an abstract class?

- a) extends**
- b) implements**
- c) abstract**
- d) new**

Answer: c

Explanation: The introduction of abstract classes was one of the major changes ushered in with PHP 5. Its inclusion in the list of new features was another sign of PHP's extended commitment to object-oriented design.

3. Which one of the following is the correct abstract method?

- a) public function write()**
- b) abstract function write()**
- c) abstract public write();**
- d) abstract public function write();**

Answer: d

Explanation: An abstract method cannot have an implementation. You declare it in the normal way, but end the declaration with a semicolon rather than a method body.

4. At least how many abstract methods must an abstract class contain?

- a) None**
- b) One**
- c) Two**
- d) Five**

Answer: b

Explanation: Classes defined as abstract may not be instantiated, and any class that contains at least one abstract method must also be abstract.

5. Which one of the following keyword is used to implement an interface?

- a) interface**
- b) get**
- c) inherit**
- d) implements**

Answer: d

Explanation: A class can implement an interface using the implements keyword in its declaration.

6. Which version of PHP introduced the concept called late static binding?

- a) PHP 4**
- b) PHP 5**
- c) PHP 5.1**
- d) PHP 5.3**

Answer: d

Explanation: None.

7. Which one of the following methods in the exception class, is used to get a nested exception object?

- a) `getPrevious()`
- b) `getCode()`
- c) `getFile()`
- d) `getLine()`

Answer: a

Explanation: `getCode()` – Get the code integer that was passed to the constructor. `getFile()` – Get the file in which the exception was generated. `getLine()` – Get the line number at which the exception was generated.

8. Which one of the following keyword is used in conjunction with an Exception object?

- a) `throws`
- b) `exception`
- c) `throw`
- d) `final`

Answer: c

Explanation: The `throw` keyword is used in conjunction with an Exception object. It halts the execution of the current method and passes responsibility for handling the error back to the calling code.

9. Which keyword is used to put a stop on inheritance?

- a) `stop`
- b) `end`
- c) `break`
- d) `final`

Answer: d

Explanation: A final class cannot be subclassed.

10. PHP provides built-in interceptor methods, which can intercept messages sent to undefined methods and properties. This is also known as _____

- a) `overloading`
- b) `overriding`
- c) `overbending`
- d) `overbinding`

Answer: a

Explanation: None.

1. Which one of the following method is invoked when a value is assigned to an undefined property?

- a) `__get()`
- b) `__set()`
- c) `__isset()`
- d) `__call()`

Answer: b

Explanation: The `__set()` method is invoked when client code attempts to assign to an undefined property. It is passed two arguments: the name of the property, and the value the client is attempting to set.

2. Which one of the following method is invoked when an undefined method is called by client code?

- a) `__get()`
- b) `__isset()`
- c) `__unset()`
- d) `__call()`

Answer: d

Explanation: The `__call()` method is probably the most useful of all the interceptor methods. The `__call()` method can be useful for delegation. Delegation is the mechanism by which one object passes method invocations on to a second.

3. Which method introduced in PHP 5, is invoked just before an object is a garbage collected?

- a) `__collect()`
- b) `__garbage()`
- c) `__destruct()`
- d) `__destructor()`

Answer: c

Explanation: You can use this method to perform any final cleaning up that might be necessary. Imagine, for example, a class that saves itself to a database when so ordered. I could use the `__destruct()` method to ensure that an instance saves its data when it is deleted.

4. Which one of the following PHP statements is true?

```
1. class CopyMe {}  
  
2. $first = new CopyMe();  
  
3. $second = $first;
```

- a) **In PHP 4: `$second` and `$first` are 2 distinct objects**
- b) **In PHP 5: `$second` and `$first` are 2 distinct objects**
- c) **In PHP 4: `$second` and `$first` refer to one object**
- d) **None of the mentioned**

Answer: a

Explanation: None.

5. Which keyword must be added before `$first` variable on the third line of the above question to make `$second` and `$first` as distinct objects in PHP 5?

- a) `copy`
- b) `clone`
- c) `cut`
- d) **Can't add any word to make them distinct**

Answer: b

Explanation: Clone operates on an object instance, producing a by-value copy.

6. What will be the output of the following PHP code? (Before the version PHP 5.2)

```
1. class StringThing {}  
  
2. $st = new StringThing();  
  
3. print $st;
```

- a) **Object Not Found**
- b) **Object id #1**
- c) **PHP Catchable fatal error**
- d) **Cannot initialize object**

Answer: b

Explanation: Since PHP 5.2, this code will produce an error like this: PHP Catchable fatal error: Object of class StringThing could not be converted to string.

7. What will be the output of the following PHP code?

```
1. class Person
```

```

2. {
3.     function getName() { return "Bob"; }
4.     function getAge() { return 44; }
5.     function __toString() {
6.         $desc = $this->getName();
7.         $desc .= " (age ".$this->getAge().") ";
8.         return $desc;
9.     }
10. }
11. $person = new Person();
12. print $person;

```

- a) Object Not Found
- b) PHP Catchable fatal error
- c) BOB (age 44)
- d) BOB

Answer: c

Explanation: By implementing a `__toString()` method, you can control how your objects represent themselves when printed. The method is invoked automatically when your object is passed to `print` or `echo`, and its return value is substituted.

8. `__clone()` is run on the ____ object.

- a) original
- b) pseudo
- c) external
- d) copied

Answer: d

Explanation: `__clone()` is run on the copied object and not the original.

9. Which method is invoked when an undefined property is accessed?

- a) `__get()`
- b) `__isset()`
- c) `__unset()`
- d) `__undefined()`

Answer: a

Explanation: None.

10. What will be the output of the following PHP code?

```

1. class Checkout
2. {
3.     final function totalize()
4.     {
5.         // calculate bill

```

```

6.     }
7. }
8.
9. class IllegalCheckout extends Checkout
10. {
11.     final function totalize()
12.     {
13.         // change bill calculation
14.     }
15. }

```

- a) PHP Fatal error: Class IllegalCheckout may not inherit from final class
- b) Value of the bill calculated
- c) PHP Fatal error: Cannot find object
- d) PHP Fatal error: Cannot override final method

Answer: d

Explanation: A final class cannot be subclassed. Less drastically, a final method cannot be overridden.

1. A package is a set of related _____

- a) Objects
- b) Classes
- c) Programs
- d) Functions

Answer: b

Explanation: A package is a set of related classes, usually grouped together in some way. Packages can be used to separate parts of a system from one another.

2. Till which version of PHP, developers were forced to name their files in a global context?

- a) PHP 4
- b) PHP 5
- c) PHP 5.2
- d) PHP 5.3

Answer: d

Explanation: If you named a class ShoppingCart, it would become instantly available across your system.

- a) i)
- b) ii)
- c) iii)
- d) i), ii) & iii)

Answer: d

Explanation: A namespace is a bucket in which you can place your classes, functions and variables.

4. Which one of the following is the correct way of declaring a namespace?

- a) namespace my;
- b) namespace my();
- c) my namespace;
- d) namespace(my);

Answer: a

Explanation: The namespace declaration must be the first statement in its file.

5. Which symbol is used to declare nested namespaces?

- a) /
- b) \
- c) .
- d) |

Answer: b

Explanation: Example – namespace com\getinstance\util;

6. Output:

```
namespace main;
com\getinstance\util\Debug::helloWorld()
PHP Fatal error: Class 'main\com\getinstance\util\Debug' not found in ...
```

Using which one of the following PHP lines will the error be removed?

- a) \com\getinstance\util\Debug::helloWorld();
- b) getinstance\util\Debug::helloWorld();
- c) main.com\getinstance\util\Debug::helloWorld();
- d) \com\getinstance\util\Debug::helloWorld();

Answer: d

Explanation: PHP is looking below the namespace main for com\getinstance\util and not finding it. That's because we are using a relative namespace here. Just as you can make absolute URLs and filepaths by starting off with a separator so you can with namespaces.

7. Which keyword can be used to fix the following PHP error?

```
namespace main;
com\getinstance\util\Debug::helloWorld()
PHP Fatal error: Class 'main\com\getinstance\util\Debug' not found in ...
```

- a) fix
- b) join
- c) use
- d) namespace

Answer: c

Explanation: Use keyword allows you to alias other namespaces within the current namespace.

Example – namespace main;

```
use com\getinstance\util;
util\Debug::helloWorld();
```

8. If I already had a Debug class in the main namespace. What will be the output of the following PHP code?

```
1. namespace main;
2. use com\getinstance\util\Debug;
3.
4. class Debug {
5.     static function helloWorld() {
6.         print "hello from main\Debug";
7.     }
```

```
8. }  
9.  
10. Debug::helloWorld();
```

- a) error
- b) hello from main
- c) hello from main\Debug
- d) debug

Answer: a

Explanation: PHP Fatal error: Cannot declare class main\Debug because the name is already in use.

9. Which one of the following statements is true for include_once() and require_once()?

- a) Both are exactly the same
- b) include_once is used for files where as require_once() is not
- c) Both Handle the errors in the same way
- d) Both do not handle the errors in the same way

Answer: d

Explanation: The only difference between the include() and require() statements lies in their handling of errors. A file invoked using require() will bring down your entire process when you meet an error. The same error encountered via a call to include() will merely generate a warning and end execution of the included file.

10. Which one of the following statements is true for require() and require_once()?

- a) They are functions
- b) They are statements
- c) They'll not work if the () is not present
- d) They can not be used to require files

Answer: b

Explanation: require() and require_once() are actually statements, not functions. This means that you can omit the brackets when using them.

1. Which function was introduced to help automate the inclusion of class files?

- a) __load()
- b) __preload()
- c) __autoload()
- d) __inload()

Answer: c

Explanation: When the PHP engine encounters an attempt to instantiate an unknown class, it invokes the __autoload() function, passing it the class name as a string. It is up to the implementer to define a strategy for locating and including the missing class file.

2. How many times can you define _____ autoload in a process?

- a) once
- b) twice
- c) thrice
- d) as many times as needed

Answer: a

Explanation: __autoload is a powerful tool, but it does have some limitations. In particular, you can only define it once in a process. If you need to change your autoload function dynamically you should look at the spl_autoload_register function, which supports that functionality.

3. Which one of the following functions will you use to check that the class exists before you work with it?

- a) class_exists()

- b) `class_exists()`
- c) `exist()`
- d) `exists_class()`

Answer: b

Explanation: The `class_exists()` function accepts a string representing the class to check for and returns a Boolean true value if the class exists and false otherwise.

4. Which one of the following will you use to check the class of an object?

- a) `class()`
- b) `_class()`
- c) `class_check()`
- d) `get_class()`

Answer: d

Explanation: This function accepts any object as an argument and returns its class name as a string.

5. PHP 4 did not support `instanceof`. Instead, which function did it provide?

- a) `is()`
- b) `get_class()`
- c) `is_a()`
- d) `is_the()`

Answer: c

Explanation: As of PHP 5.3 `is_a()` no longer deprecated.

6. You use the `get_class_methods()` function to return the names of all the methods in the class. Which function will you use to print it on the screen?

- a) `printf()`
- b) `print_ar`
- c) `print_r`
- d) `echo`

Answer: c

Explanation: The function `get_class_methods` returns an array therefore we cannot use `echo`.

7. If you call a method and it doesn't exist it'll cause a problem. To check the method which function will you use?

- a) `_method()`
- b) `methodexists()`
- c) `is_callable()`
- d) `is_method()`

Answer: c

Explanation: There is another method which you can use – `method_exists()` for similar purpose.

8. What will be the output if a protected method is given as the argument to the function `method_exists()`?

- a) Method does not exist
- b) False
- c) Error
- d) True

Answer: d

Explanation: `method_exists()` returns true for private and protected methods as well as for public ones. Remember that the fact that a method exists does not mean that it will be callable.

9. Which one of the following function should I use to find the parent class of a class?

- a) `get_parent_class()`
- b) `parent_class()`

- c) `class_parent()`
- d) `get_class_parent()`

Answer: a

Explanation: The class functions also allow us to chart inheritance relationships. This function requires either an object or a class name, and it returns the name of the superclass, if any. If no such class exists, that is, if the class we are testing does not have a parent, then the function returns false.

10. Which class accepts a class name or an object reference and returns an array of interface name?

- a) `class_implements()`
- b) `is_subclass()`
- c) `is_subclass_of()`
- d) `class_interface()`

Answer: a

Explanation: `is_subclass_of()` will tell you only about class inheritance relationships. It will not tell you that a class implements an interface.

1. Object-oriented code tries to minimize dependencies by moving responsibility for handling tasks away from ____ and toward the objects in the system.

- a) server code
- b) client code
- c) machine code
- d) procedural code

Answer: b

Explanation: Procedural code takes the form of a sequential series of commands and method calls. The controlling code tends to take responsibility for handling differing conditions. This top-down control can result in the development of duplications and dependencies across a project. Object-oriented code tries to minimize these dependencies by moving responsibility for handling tasks away from client code and toward the objects in the system.

2. Placing a static method for generating _____ objects in the _____ class is convenient.

- a) child parent
- b) parent child
- c) final static
- d) static final

Answer: a

Explanation: Such a design decision does have its own consequences, however.

3. The extent to which proximate procedures are related to one another is known as _____

- a) Coupling
- b) Balancing
- c) Cohesion
- d) Co-relation

Answer: c

Explanation: Ideally, you should create components that share a clear responsibility. If your code spreads related routines widely, you will find them harder to maintain as you have to hunt around to make changes. Our Param Handler classes collect related procedures into a common context.

4. _____ occurs when discrete parts of a system's code are tightly bound up with one another so that a change in one part necessitates changes in the others.

- a) Loose Coupling
- b) Tight Coupling
- c) Co-relation
- d) Balancing

Answer: b

Explanation: Tight coupling is by no means unique to procedural code, though the sequential nature of such code makes it prone to the problem.

5. _____ code makes change easier because the impact of altering an implementation will be localized to the component being altered.

- a) Orthogonal**
- b) Cohesion**
- c) Coupling**
- d) Balancing**

Answer: a

Explanation: Orthogonality, it is argued, promotes reuse in that components can be plugged into new systems without needing any special configuration. Such components will have clear inputs and outputs independent of any wider context.

6. Polymorphism is also known as _____

- a) switch**
- b) overact**
- c) encapsulation**
- d) class switching**

Answer: d

Explanation: Polymorphism is the maintenance of multiple implementations behind a common interface.

7. Which one of the following is known as the key to object-oriented programming?

- a) Polymorphism**
- b) Encapsulation**
- c) Data Abstraction**
- d) Orthogonality**

Answer: b

Explanation: Encapsulation simply means the hiding of data and functionality from a client.

8. Which one among the following means tight coupling?

- a) Code Duplication**
- b) Inheritance**
- c) Encapsulation**
- d) Polymorphism**

Answer: a

Explanation: Duplication is one of the great evils in code. Take a look at the instances of repetition in your system. Perhaps they belong together. If you change something fundamental about one routine, will the similar routines need amendment? If this is the case, they probably belong in the same class.

9. UML stands for?

- a) unified mailing language**
- b) unified modeling logo**
- c) undefined modeling language**
- d) unified modeling language**

Answer: d

Explanation: The UML emerged as a standard only after long years of intellectual and bureaucratic sparring among the great and good of the object oriented design community.

10. In a class diagram the class is divided into three sections, what is displayed in the first section?

- a) Class Attributes**
- b) Class Declaration**

- c) Class Name**
- d) Class Functions**

Answer: c

Explanation: The other two sections are optional when no more information than the class name is known.

- 1. _____ are used in class diagrams to describe the way in which specific elements should be used.**
- a) Attributes**
 - b) Constraints**
 - c) Constants**
 - d) Class Names**

Answer: b

Explanation: The {abstract} syntax is an example of a constraint. There is no special structure for the text between the braces; it should simply provide a short clarification of any conditions that may apply to the element.

- 2. Which one of the following is displayed below the class name in the class diagrams?**
- a) Functions**
 - b) Methods**
 - c) Attributes**
 - d) Constraints**

Answer: c

Explanation: Attributes describe a class's properties. Attributes are listed in the section directly beneath the class name.

- 3. + is the visibility code for?**
- a) Public**
 - b) Private**
 - c) Protected**
 - d) Friendly**

Answer: a

Explanation: The visibility symbol is followed by the name of the attribute.

- 4. Which relationship is illustrated by a line that begins with an unfilled diamond?**
- a) Composition**
 - b) Abstraction**
 - c) Aggregation**
 - d) Inheritance**

Answer: c

Explanation: In the case of aggregation, the contained objects are a core part of the container, but they can also be contained by other objects at the same time.

- 5. If the diamond is filled it depicts which relationship?**
- a) Strong Aggregation**
 - b) Composition**
 - c) Abstraction**
 - d) Inheritance**

Answer: b

Explanation: In composition, the contained object can be referenced by its container only. It should be deleted when the container is deleted.

- 6. Which one of the following statements is true about sequence diagrams?**
- a) A sequence diagram is class based**
 - b) A sequence diagram presents the participants of a system from right to left**
 - c) The vertical broken lines represent the lifetime of the class in the system**

d) A sequence diagram is object based

Answer: d

Explanation: A sequence diagram is object based rather than class based. It is used to model a process in a system step by step.

7. A bidirectional relationship in class diagrams is described by_____

- a) double-headed arrow**
- b) visibility symbols**
- c) single-headed arrow**
- d) double diamond**

Answer: a

Explanation: A unidirectional association has single-headed arrow.

8. # is the visibility code for?

- a) Private**
- b) Friendly**
- c) Protected**
- d) Static**

Answer: c

Explanation: Available to the current class and its subclasses only.

9. Which one of the following is displayed in the third section of the class diagram?

- a) Operations**
- b) Inheritance**
- c) Abstraction**
- d) Coupling**

Answer: a

Explanation: Operations describe methods, or more properly, they describe the calls that can be made on an instance of a class.

10. Inheritance in class diagrams is depicted by_____

- a) single-headed empty arrow**
- b) single-headed filled arrow**
- c) double-headed empty arrow**
- d) double-headed filled arrow**

Answer: a

Explanation: The UML describes the inheritance relationship as a generalization. This relationship is signified by a line leading from the subclass to its parent. The line is tipped with an empty closed arrow.

1. What will be the output of the following PHP code?

```
1. <?php
2. "Hello World"
3. ?>
```

- a) Error**
- b) Hello World**
- c) Nothing**
- d) Missing semicolon error**

Answer: c

Explanation: If you need to output something onto the screen you'll need to use echo or print_r.

2. What will be the output of the following PHP code?

```
1. <?php
2. print_r "Hello world"
3. ?>
```

- a) Error
- b) Hello World
- c) Nothing
- d) Missing semicolon error

Answer: a

Explanation: The statement should be `print_r('Hello World')` to print Hello world. Also if there is only one line then there is no requirement of a semicolon, but it is better to use it.

3. What will be the output of the following PHP code?

```
1. <?php
2. echo 'Hello World';
3. <html>
4. Hello world
5. </html>
6. ?>
```

Answer: d

Explanation: Parse error: syntax error, unexpected '<' on line 2. You can not use the html tag inside php tags.

4. What will be the output of the following PHP code?

```
1. <?php
2. Echo "Hello World1";
3. echo " Hello world2";
4. ECHO " Hello world3";
5. ?>
```

Answer: a

Explanation: In PHP, all user-defined functions, classes, and keywords (e.g. if, else, while, echo, etc.) are case-insensitive.

5. What will be the output of the following PHP code?

```
1. <?php
2. $color = "red";
3. echo "$color";
4. echo "$COLOR";
5. echo "$Color";
6. ?>
```

- a) redredred

- b) redred
- c) red
- d) Error

Answer: c

Explanation: In PHP, all variables are case-sensitive.

6. What will be the output of the following PHP code?

```
1. <?php
2.  # echo "Hello world";
3.  echo "# Hello world";
4. ?>
```

- a) # Hello world
- b) Hello world# Hello world
- c) Hello world
- d) Error

Answer: a

Explanation: # is a single line comment.

7. What will be the output of the following PHP code?

```
1. <?php
2. echo "<i>Hello World</i>"
3. ?>
```

- a) Hello world
- b) Hello world in italics
- c) Nothing
- d) Error

Answer: b

Explanation: You can use tags like italics, bold etc. inside php script.

8. What will be the output of the following PHP code?

```
1. <?php
2. echo "echo "Hello World"";
3. ?>
```

- a) Hello world
- b) echo “Hello world”
- c) echo Hello world
- d) Error

Answer: d

Explanation: It would have printed echo “Hello world” if we have put backslash doublequotes just before and just after Hello World string.

9. What will be the output of the following PHP code?

```
1. <?php
```

```
2. <?php
3. echo "Hello world";
4. ?>
5. ?>
```

- a) HELLO WORLD
- b) Hello world
- c) Nothing
- d) Error

Answer: d

Explanation: You can not have php tags inside a php tag.

10. What will be the output of the following PHP code?

```
1. <?php
2. $color = red;
3. echo "\$color";
4. ?>
```

- a) red
- b) \$color
- c) \red
- d) Error

Answer: b

Explanation: To print red remove the \.

1. What will be the output of the following PHP code?

```
1. <?php
2. /*
3. echo "Hello world";
4. */
5. ?>
```

2. What will be the output of the following PHP code?

```
1. <?php
2. $color = red;
3. echo "$color" . red ;
4. ?>
```

- a) red red
- b) red
- c) error
- d) nothing

Answer: c

Explanation: Use of undefined constant red.

3. What will be the output of the following PHP code?

```
1. <?php
2. $color1 = red;
3. $color2 = green;
4. echo "$color1"."$color2";
5. ?>
```

- a) red green
- b) red
- c) green
- d) error

Answer: d

Explanation: It has to be \$color1 = "red"; and \$color2 = "green"; therefore the error:

4. What will be the output of the following PHP code?

```
1. <?php
2. $color = "red";
3. $color = "green";
4. echo "$color";
5. ?>
```

- a) red
- b) green
- c) red green
- d) error

Answer: b

Explanation: The variable contains the last value which has been assigned.

5. What will be the output of the following PHP code?

```
1. <?php
2. $color1 = "red";
3. $color2 = "green";
4. echo "$color1" . "$color2";
5. ?>
```

- a) red
- b) green
- c) red green
- d) redgreen

Answer: d

Explanation: The . operator is used to join to strings.

6. What will be the output of the following PHP code?


```
1. <?php
2. $color1 = "red";
3. $color2 = "green";
4. echo "$color1" + "$color2";
5. ?>
```

- a) redgreen
- b) red green
- c) 0
- d) error

Answer: c

Explanation: + operator does not join both the strings.

7. What will be the output of the following PHP code?

```
1. <?php
2. $color1 = "red";
3. $color2 = "red";
4. echo "$color1" + "$color2";
5. ?>
```

- a) redgreen
- b) red green
- c) 0
- d) 1

Answer: c

Explanation: + does not return 1 if the variables are equal.

8. What will be the output of the following PHP code?

```
1. <?php
2. $color1 = "red";
3. $color2 = "1";
4. echo "$color1" + "$color2";
5. ?>
```

- a) red1
- b) red 1
- c) 0
- d) 1

Answer: d

Explanation: + just returns the numeric value even though it is inside double quotes.

9. What will be the output of the following PHP code?

```
1. <?php
2. $color1 = "1";
```

```
3. $color2 = "1";  
4. echo "$color1" + "$color2";  
5. ?>
```

- a) 11
- b) 2
- c) 0
- d) 1

Answer: b

Explanation: + can be used to add to integer values which are enclosed by double-quotes.

10. What will be the output of the following PHP code?

```
1. <?php  
2. $color1 = "red";  
3. $color2 = "1";  
4. $color3 = "grey"  
5. echo "$color1" + "$color2" . "$color3";  
6. ?>
```

- a) 1grey
- b) grey
- c) 0
- d) red1grey

Answer: a

Explanation: + gives the value 1 and . is used to give join 1 and grey.

1. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 5;  
3. $y = 10;  
4. $z = "$x + $y";  
5. echo "$z";  
6. ?>
```

- a) 15
- b) 10 + 5
- c) \$z
- d) \$x + \$y

Answer: b

Explanation: Variable z will store 10 + 5 because 10 + 5 is given in double-quotes.

2. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 4;
```

```
3. $y = 3;  
4. $z = 1;  
5. echo "$x = $x + $y + $z";  
6. ?>
```

- a) $4 = 4 + 3 + 1$
- b) 8
- c) $8 = 4 + 3 + 1$
- d) Error

Answer: a

Explanation: Again since the variables are inside double quotes we get this result.

3. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 4;  
3. $y = 3  
4. $z = 1;  
5. $z = $z + $x + $y;  
6. echo "$z";  
7. ?>
```

- a) \$z
- b) 15
- c) 8
- d) 1

Answer: c

Explanation: Normal addition of variables x, y and z occurs and result of 8 will be displayed.

4. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 3.3;  
3. $y = 2;  
4. echo $x % $y;  
5. ?>
```

- a) 0
- b) 1
- c) 2
- d) Error

Answer: b

Explanation: % is the modulo operator. Unlike in C we can use it get reminder or floating point numbers in PHP.

5. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $x = 10;  
3. $y = 4;  
4. $z = 3;  
5. echo $x % $y % $z;  
6. ?>
```

- a) 0
- b) 1
- c) 2
- d) Error

Answer: c

Explanation: The expression is considered as $(\$x\%\$y)\%z$ in this case $(10\%4)\%3$ which is 2.

6. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 10;  
3. $y = 4;  
4. $z = 3;  
5. echo ($x % ($y) + $z);  
6. ?>
```

- a) 5
- b) 3
- c) 0
- d) 1

Answer: a

Explanation: The innermost bracket is evaluated first, since it covers only variable y it is as good as not using brackets.

7. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 30;  
3. $y = 20;  
4. $z = 10;  
5. echo $x + $y - $z / ($z - $y);  
6. ?>
```

- a) 41
- b) -4
- c) -5
- d) 51

Answer: d

Explanation: First $(\$z - \$y)$ is evaluated then $-\$z/(\$z - \$y)$ is evaluated this results in 1 which is added to $\$x + \y therefore we get 51.

8. What will be the output of the following PHP code?

```
1. <?php
2. $x = -1;
3. $y = 1;
4. $z = $x * $y + $z;
5. echo $z;
6. ?>
```

Answer: c

Explanation: Since the variable z is not defined it returns the error also it takes z as 0 and returns the value -1.

9. What will be the output of the following PHP code?

```
1. <?php
2. $x = 4;
3. $y = -3;
4. $z = 11;
5. echo 4 + $y * $z / $x;
6. ?>
```

- a) 4.25
- b) 3.25
- c) -3.25
- d) -4.25

Answer: d

*Explanation: First the * is evaluated then / followed by + therefore we can rewrite this expression as $4 + ((-3 * 11) / 4)$ which results in -4.25.*

10. What will be the output of the following PHP code?

```
1. <?php
2. $x = 3.5;
3. $y = 2;
4. $z = 2;
5. echo $x / $y / $z;
6. ?>
```

- a) 1.75
- b) 0.875
- c) 3.5
- d) Error

Answer: b

Explanation: First $\$x / \y is evaluated then this is divided by $\$z$ therefore we get 0.875.

1. What will be the output of the following PHP code?

```
1. <?php
2. one = 1;
3. two = 2;
4. three = 3;
5. four = 4;
6. echo "one / two + three / four";
7. ?>
```

- a) 0.75
- b) 0.05
- c) 1.25
- d) Error

Answer: d

Explanation: Variables should start with a \$ symbol, since one, two, three, four don't begin with \$ symbol we'll get an error.

2. What will be the output of the following PHP code?

```
1. <?php
2. $on$e = 1;
3. $tw$o = 2;
4. $thre$e = 3;
5. $fou$r = 4;
6. echo "$on$e / $tw$o + $thre$e / $fou$r";
7. ?>
```

- a) 0.75
- b) 0.05
- c) 1.25
- d) Error

Answer: d

Explanation: You can not use the \$ in between the variable name.

3. What will be the output of the following PHP code?

```
1. <?php
2. $on_e = 1;
3. $tw_o = 2;
4. $thre_e = 3;
5. $fou_r = 4;
6. echo $on_e / $tw_o + $thre_e / $fou_r;
7. ?>
```

- a) 0.75

- b) 0.05
- c) 1.25
- d) Error

Answer: c

Explanation: You can use _ in a variable name.

4. What will be the output of the following PHP code?

```
1. <?php
2. $On_e = 1;
3. $tw_o = 2;
4. $thre_e = 3;
5. $fou_r = 4;
6. echo $on_e / $tw_o + $thre_e / $fou_r;
7. ?>
```

- a) 0.75
- b) 0.05
- c) 1.25
- d) Error

Answer: a

Explanation: Since the variable initialised is \$On_e and the variable in the echo statement is \$on_e the echo statement treats \$on_e as 0;

5. What will be the output of the following PHP code?

```
1. <?php
2. echo $red;
3. ?>
```

- a) 0
- b) Nothing
- c) True
- d) Error

Answer: b

Explanation: There will no output returned as the variable \$red does not hold any value.

6. What will be the output of the following PHP code?

```
1. <?php
2. $four4 = 4;
3. $three3 = 3;
4. $two2 = 2;
5. echo $four4 + $three3 / $two2 - 1;
6. ?>
```

- a) 4.5

- b) 7
- c) 3.5
- d) Error

Answer: a

Explanation: You can use numbers in a variable name.

7. What will be the output of the following PHP code?

```
1. <?php
2. $4four = 4;
3. $3three = 3;
4. $2two = 2;
5. echo $4four + $3three / $2two - 1;
6. ?>
```

- a) 4.5
- b) 7
- c) 3.5
- d) Error

Answer: d

Explanation: A variable name can not start with a numeric value.

8. What will be the output of the following PHP code?

```
1. <?php
2. int $one = 1;
3. echo "$one";
4. ?>
```

- a) 0
- b) 1
- c) \$one
- d) Error

Answer: d

Explanation: Unlike other programming languages there are no data types in PHP.

9. What will be the output of the following PHP code?

```
1. <?php
2. var $one = 1;
3. var $two = 2;
4. echo $one / $two * $one / $two * $two;
5. ?>
```

- a) 1
- b) 0
- c) 0.5

d) Error

Answer: d

Explanation: You can not use var before a variable name.

10. What will be the output of the following PHP code?

```
1. <?php
2. $hello = "Hello World";
3. $bye = "Bye";
4. echo $hello;"$bye";
5. ?>
```

a) Hello World

b) Bye

c) Hello worldBye

d) Error

Answer: a

Explanation: Since there is a semi-colon in between \$hello and \$bye, the line ends at \$hello. However \$bye would have printed if a echo was present before "\$bye".

1. What will be the output of the following PHP code?

```
1. <?php
2. $x;
3. echo "$x";
4. ?>
```

a) 0

b) 1

c) Nothing

d) Error

Answer: c

Explanation: Since the variable x is not initialised it is not storing any value, therefore nothing will be printed on the screen.

2. What will be the output of the following PHP code?

```
1. <?php
2. $x = 5;
3. {
4.     $x = 10;
5.     echo "$x";
6. }
7. echo "$x";
8. ?>
```

- a) 1010
- b) 105
- c) 510
- d) error

Answer: a

Explanation: Variable x stores the value 10 and not 5.

3. What will be the output of the following PHP code?

```
1. <?php
2. $x = 5;
3. {
4.     echo "$x";
5. }
6. ?>
```

- a) 0
- b) 5
- c) Nothing
- d) Error

Answer: b

Explanation: The variable x stores the value 5 and therefore the value 5 is printed on the screen.

4. What will be the output of the following PHP code?

```
1. <?php
2. $x = 5;
3. function fun()
4. {
5.     echo "$x";
6. }
7. fun();
8. ?>
```

- a) 0
- b) 5
- c) Nothing
- d) Error

Answer: c

Explanation: The variable x is not defined inside the function fun(), therefore nothing is printed on the screen.

5. What will be the output of the following PHP code?

```
1. <?php
2. $x = 5;
3. function fun()
```

```
4. {  
5.     $x = 10;  
6.     echo "$x";  
7. }  
8. fun();  
9. echo "$x";  
10. ?>
```

- a) 0
- b) 105
- c) 510
- d) Error

Answer: b
Explanation: First when the function is called variable x is initialised to 10 so 10 is printed later the global value 5 is printed.

6. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 4;  
3. $y = 3;  
4. function fun($x = 3, $y = 4)  
5. {  
6.     $z = $x+$y/$y+$x;  
7.     echo "$z";  
8. }  
9. echo $x;  
10. echo $y;  
11. echo $z;  
12. fun($x, $y);  
13. ?>
```

- a) 43
- b) 943
- c) 349
- d) 439

Answer: d
Explanation: Firstly, the statements outside the function are printed, since z is not defined it'll no value is printed for z. Next the function is called and the value of z inside the function is printed.

7. What will be the output of the following PHP code?

```
1. <?php
```

```

2. $x = 4;

3. $y = 3;

4. function fun($x, $y)

5. {

6.     $z = $x + $y / $y + $x;

7.     echo "$z";

8. }

9. echo $x;

10. echo $y;

11. echo $z;

12. fun(3, 4);

13. ?>

```

- a) 437
- b) 439
- c) 349
- d) 347

Answer: a

Explanation: It is same as above but the value passed into the function is 3,4 and not 4,3. Therefore the difference in answer.

8. What will be the output of the following PHP code?

```

1. <?php

2. function fun($x,$y)

3. {

4.     $x = 4;

5.     $y = 3;

6.     $z = $x + $y / $y + $x;

7.     echo "$z";

8. }

9. fun(3, 4);

10. ?>

```

- a) 7
- b) 9
- c) 0
- d) Error

Answer: b

Explanation: Value 3, 4 is passed to the function but that is lost because x and y are initialised to 4 and 3 inside the function. Therefore we get the given result.

9. What will be the output of the following PHP code?

```
1. <?php
2. $x = 3, 4, 5, 6;
3. echo "$x";
4. ?>
```

- a) 3
- b) 4
- c) 6
- d) Error

Answer: d

Explanation: In C you won't get an error but in PHP you'll get a syntax error.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a = 10;
3. $b = 4;
4. $c = fun(10,4);
5. function fun($a,$b)
6. {
7.     $b = 3;
8.     return $a - $b + $b - $a;
9. }
10. echo $a;
11. echo $b;
12. echo $c;
13. ?>
```

- a) 104
- b) 410
- c) 1400
- d) 4100

Answer: c

Explanation: The value returned from the function is 0, and value of a is 10, value of b is 4 and c is 0.

1. What will be the output of the following PHP code?

```
1. <?php
2. $a = "$winner";
3. $b = "/$looser";
4. echo $a,$b;
```

5. ?>

- a) \$winner/\$looser
- b) /\$looser
- c) /
- d) \$looser

Answer: c

Explanation: Since variables \$winner and \$looser is not defined we only see / as output.

2. What will be the output of the following PHP code?

```
1. <?php
2. $a = "$winner";
3. $b = "\$looser";
4. echo $a, $b;
5. ?>
```

- a) \$winner\$looser
- b) \ \$looser
- c) \
- d) \$looser

Answer: d

Explanation: As there is a backslash before \$ it takes it as a string and not a variable therefore we get \$looser as the output.

3. What will be the output of the following PHP code?

```
1. <?php
2. $a = "$winner";
3. $b = "\\ $looser";
4. echo $a, $b;
5. ?>
```

- a) \$winner\\ \$looser
- b) \\ \$looser
- c) \
- d) \$looser

Answer: c

Explanation: Since two backslashes are used together, a single backslash is printed on the screen and as \$looser is not initialised only single backslash is printed.

4. What will be the output of the following PHP code?

```
1. <?php
2. $x = 5;
3. $y = 10;
4. function fun()
```

```
5. {  
6.     $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];  
7. }  
8. fun();  
9. echo $y;  
10. ?>
```

- a) 5
- b) 10
- c) 15
- d) Error

Answer: c

Explanation: You can access the global variable using \$GLOBALS['globalvariablename'].

5. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 5;  
3. $y = 10;  
4. function fun()  
5. {  
6.     $y = $GLOBALS['x'] + $GLOBALS['y'];  
7. }  
8. fun();  
9. echo $y;  
10. ?>
```

- a) 5
- b) 10
- c) 15
- d) Error

Answer: b

Explanation: The value of global variable y does not change therefore it'll print 10;

6. What will be the output of the following PHP code?

```
1. <?php  
2. function fun()  
3. {  
4.     $x = 0;  
5.     echo $x;  
6.     $x++;
```

```
7. }  
8. fun();  
9. fun();  
10. fun();  
11. ?>
```

- a) 012
- b) 123
- c) 000
- d) 111

Answer: c

Explanation: Every time the function is called the value of x becomes 0, therefore we get 0 on every function call.

7. What will be the output of the following PHP code?

```
1. <?php  
2. function fun()  
3. {  
4.     static $x = 0;  
5.     echo $x;  
6.     $x++;  
7. }  
8. fun();  
9. fun();  
10. fun();  
11. ?>
```

- a) 012
- b) 123
- c) 111
- d) Error

Answer: a

Explanation: When static is used, each time the function is called, that variable will still have the information it contained from the last time the function was called.

8. What will be the output of the following PHP code?

```
1. <?php  
2. static $x = 0;  
3. function fun()  
4. {  
5.     echo $x;  
6.     $x++;
```



```
7. }  
8. fun();  
9. fun();  
10. fun();  
11. ?>
```

- a) 012
- b) 123
- c) Nothing
- d) Error

Answer: c

Explanation: Since variable x is not defined inside the function fun(), nothing will be printed.

9. What will be the output of the following PHP code?

```
1. <?php  
2. $x=0;  
3. function fun()  
4. {  
5.     echo $GLOBALS['x'];  
6.     $GLOBALS['x']++;  
7. }  
8. fun();  
9. fun();  
10. fun();  
11. ?>
```

- a) 000
- b) 012
- c) 123
- d) Error

Answer: b

Explanation: Since, we are using \$GLOBALS['x'].

10. What will be the output of the following PHP code?

```
1. <?php  
2. $x = 0;  
3. function fun()  
4. {  
5.     echo $GLOBALS['x'];  
6.     $x++;
```

```
7. }  
8. fun();  
9. fun();  
10. fun();  
11. ?>
```

- a) 000
- b) 012
- c) Nothing
- d) Error

Answer: a

Explanation: Every time the function is called the value of x becomes 0, therefore we get 0 on every function call.

1. What will be the output of the following PHP code?

```
1. <?php  
2. echo "Hello world </br> I am learning PHP";  
3. ?>
```

Answer: c

Explanation: The break tag is used as break line.

2. What will be the output of the following PHP code?

```
1. <?php  
2. echo "Hello world <strong>I am learning PHP</strong>"  
3. ?>
```

Answer: b

Explanation: None.

3. What will be the output of the following PHP code?

```
1. <?php  
2. echo "This", "was", "a", "bad", "idea";  
3. ?>
```

- a) This, was, a, bad, idea
- b) This was a bad idea
- c) Thiswasabadidea
- d) Error

Answer: c

Explanation: In an echo statement the comma operator is used to join strings.

4. What will be the output of the following PHP code?

```
1. <?php  
2. echo "This"."was"."a"."bad"."idea";  
3. ?>
```

- a) This, was, a, bad, idea
- b) This was a bad idea
- c) Thiswasabadidea
- d) Error

Answer: c

Explanation: In an echo statement the dot operator is used to join strings.

5. What will be the output of the following PHP code?

```
1. <?php
2. echo "This", "was"|"a", "bad"."idea";
3. ?>
```

- a) Thiswasabadidea
- b) Thiswasbadidea
- c) Thiswas a badidea
- d) Thiswas abadidea

Answer: b

Explanation: You can use only comma and dot operator to join starings, other characters do not have the same function.

6. What will be the output of the following PHP code?

```
1. <?php
2. $one = "Hello";
3. $two = "World";
4. echo $one, $two;
5. ?>
```

- a) Hello World
- b) Hello
- c) World
- d) HelloWorld

Answer: d

Explanation: Echo can print two variables which are separated by a comma.

7. What will be the output of the following PHP code?

```
1. <?php
2. $one = "Hello";
3. $two = "World";
4. echo "$one$two";
5. ?>
```

- a) HelloWorld
- b) \$one\$two
- c) Hello
- d) Error

Answer: a

Explanation: Even though both the variables are inside the double-quotes, the value of the variable is substituted and then printed to the screen.

8. What will be the output of the following PHP code?

```
1. <?php
2. $one = "Hello";
3. $two = "World";
4. echo "$one"+"$two";
5. ?>
```

- a) HelloWorld
- b) Hello+World
- c) 0
- d) Error

Answer: c
Explanation: In an echo statement you can not use plus sign to join two strings.

9. What will be the output of the following PHP code?

```
1. <?php
2. echo "This is <i>India</i>";
3. ?>
```

- a) This is India
- b) This is *India*
- c) This is
- d) Error

Answer: b
Explanation: None.

10. What will be the output of the following PHP code?

```
1. <?php
2. $cars = array("Volvo", "BMW", "Toyota");
3. echo "My car is a {$cars[0]}";
4. ?>
```

- a) My car is a Volvo
- b) My car is a BMW
- c) My car is a Toyota
- d) Error

Answer: a
Explanation: In the echo statement the {\$cars[0]} is replaced by the 1st element in cars that is Volvo.

1. What will be the output of the following PHP code?

```
1. <?php
2. print "echo hello world";
```

3. ?>

- a) echo hello world
- b) hello world
- c) nothing
- d) error

Answer: a

Explanation: The print statement will print whatever is present inside the double-quotes.

2. What will be the output of the following PHP code?

```
1. <?php
2. $one = 1;
3. print($one);
4. print $one;
5. ?>
```

- a) 01
- b) 11
- c) 10
- d) Error

Answer: b

Explanation: Print can be used with or without parentheses.

3. What will be the output of the following PHP code?

```
1. <?php
2. $cars = array("Volvo", "BMW", "Toyota");
3. print $cars[2];
4. ?>
```

- a) Volvo
- b) BMW
- c) Toyota
- d) Error

Answer: c

Explanation: Print statement can be used to output a specific array member.

4. What will be the output of the following PHP code?

```
1. <?php
2. $one = "one";
3. $two = "two";
4. print($one$two);
5. ?>
```

- a) onetwo
- b) one

- c) nothing
- d) error

Answer: d

Explanation: The above syntax will produce an error, unlike the echo statement.

5. What will be the output of the following PHP code?

```
1. <?php
2. $one = "one";
3. $two = "two";
4. print($one,$two);
5. ?>
```

- a) onetwo
- b) one, two
- c) one
- d) error

Answer: d

Explanation: The above syntax will produce an error, unlike the echo statement.

6. What will be the output of the following PHP code?

```
1. <?php
2. $one = "one";
3. $two = "two";
4. print("$one$two");
5. ?>
```

- a) onetwo
- b) \$one\$two
- c) one
- d) error

Answer: a

Explanation: This is same as the echo statement.

7. What will be the output of the following PHP code?

```
1. <?php
2. $one = "one";
3. $two = "two";
4. print($one==$two);
5. ?>
```

- a) true
- b) false
- c) nothing
- d) error

Answer: c

Explanation: Since we are equating two unequal strings we do not get any output.

8. What will be the output of the following PHP code?

```
1. <?php
2. $one = "one";
3. $two = "one";
4. print($one == $two);
5. ?>
```

- a) true
- b) false
- c) 1
- d) error

Answer: c

Explanation: Since both the strings are equal the result 1 is printed on the screen.

9. What will be the output of the following PHP code?

```
1. <?php
2. print "Hello world!<br>";
3. print "I'm about to learn PHP!";
4. ?>
```

Answer: c

Explanation: Most of the properties of echo and print are same. Strings can contain HTML markup.

10. What will be the output of the following PHP code?

```
1. <?php
2. print("this"."was"."a"."bad"."idea");
3. ?>
```

- a) thiswasabadidea
- b) this was a bad idea
- c) nothing
- d) error

Answer: a

Explanation: You can use the dot operator like in echo but you can not use the comma operator to do the same.

1. What will be the output of the following PHP code?

```
1. <?php
2. define("GREETING", "PHP is a scripting language", true);
3. echo GREETING;
4. echo "<br>"
5. echo GREETING;
```

6. ?>

2. What will be the output of the following PHP code?

```
1. <?php
2. define("GREETING", "PHP is a scripting language");
3. echo $GREETING;
4. ?>
```

- a) \$GREETING
- b) no output
- c) PHP is a scripting language
- d) GREETING

Answer: b

Explanation: Constants do not need a \$ before them, they are referenced by their variable names itself.

3. What will be the output of the following PHP code?

```
1. <?php
2. define('GREETING_TEST', 'PHP is a scripting language', true);
3. echo GREETING_TEST;
4. $changing_variable = 'test';
5. echo constant('GREETING_' . strtoupper($changing_variable));
6. ?>
```

4. What will be the output of the following PHP code?

```
1. <?php
2. class Constants
3. {
4.     define('MIN_VALUE', '0.0');
5.     define('MAX_VALUE', '1.0');
6.     public static function getMinValue()
7.     {
8.         return self::MIN_VALUE;
9.     }
10.    public static function getMaxValue()
11.    {
12.        return self::MAX_VALUE;
13.    }
14. }
15. echo Constants::getMinValue();
```



```
16. echo Constants::getMaxValue();
```

```
17. ?>
```

- a) 0.01.0
- b) 01
- c) No output
- d) ERROR

Answer: d

Explanation: In a class constants should be defined `const MIN_VALUE = 0.0;const MAX_VALUE = 1.0;` instead.

5. What will be the output of the following PHP code?

```
1. <?php
2. define("__LINE__", "PHP is a scripting language");
3. echo __LINE__;
4. ?>
```

- a) PHP is a scripting language
- b) __LINE__
- c) 2
- d) ERROR

Answer: c

Explanation: __LINE__ is a magical constant that gives the current line number and cannot be used a variable/constant name.

6. What will be the output of the following PHP code?

```
1. <?php
2. define('IF', 42);
3. echo "IF: ", IF;
4. ?>
```

- a) IF:42
- b) No output
- c) IF:
- d) ERROR

Answer: d

Explanation: Keyword like IF cannot be used as constant names.

7. What will be the output of the following PHP code?

```
1. <?php
2. define("NEW_GOOD_NAME_CONSTANT", "I have a value");
3. define("OLD_BAD_NAME_CONSTANT", NEW_GOOD_NAME_CONSTANT);
4.
5. echo NEW_GOOD_NAME_CONSTANT;
6. echo OLD_BAD_NAME_CONSTANT;
```

7. ?>

- a) I have a value
- b) I have a valueI have a value
- c) ERROR
- d) I have a valueNEW_GOO_NAME_CONSTANTS

Answer: b

Explanation: Constants can be set as values for other constants.

8. What will be the output of the following PHP code?

```
1. <?php
2. define('GOOD_OCTAL', 0700);
3. define('BAD_OCTAL', 0600);
4. print GOOD_OCTAL;
5. print '<br>';
6. print BAD_OCTAL;
7. ?>
```

Answer: a

Explanation: Anything starting from 0 is evaluated as an octal.

9. What will be the output of the following PHP code?

```
1. <?php
2. define("VAR_NAME", "test");
3. ${VAR_NAME} = "value";
4. echo VAR_NAME;
5. echo ${VAR_NAME};
6. ?>
```

- a) test
- b) testtest
- c) testvalue
- d) error; constant value cannot be changed

Answer: c

Explanation: `${VAR_NAME}` creates a new variable which is not same as `VAR_NAME`.

10. What will be the output of the following PHP code?

```
1. <?php
2. class myObject { }
3. define('myObject::CONSTANT', 'test');
4. echo myObject::CONSTANT;
5. ?>
```

- a) test
- b) error
- c) myObject::CONSTANT
- d) no output

Answer: b

Explanation: Class constants cannot be defined outside class.

1. What will be the output of the following PHP code?

```
1. <?php
2. $a = 10;
3. echo ++$a;
4. echo $a++;
5. echo $a;
6. echo ++$a;
7. ?>
```

- a) 11111213
- b) 11121213
- c) 11111212
- d) 11111112

Answer: a

Explanation: ++\$a increments a and then prints it, \$a++ prints and then increments.

2. What will be the output of the following PHP code?

```
1. <?php
2. $a = 12;
3. --$a;
4. echo $a++;
5. ?>
```

- a) 11
- b) 12
- c) 10
- d) error

Answer: a

Explanation: The + operator does union of arrays in that order, then the === operator compares key and value pairs.

3. What will be the output of the following PHP code?

```
1. <?php
2. $x = "test";
3. $y = "this";
4. $z = "also";
5. $x .= $y .= $z ;
```

6. echo \$x;

7. echo \$y;

8. ?>

- a) testthisalso
- b) testthis
- c) testthisalsothisalso
- d) error at line 4

Answer: c

Explanation: The $x .= y$ is a shorthand for $x = x.y$ and this is evaluated from right to left.

4. What will be the output of the following PHP code?

```
1. <?php
2. $x = 1;
3. $y = 2;
4. if (++$x == $y++)
5. {
6.     echo "true ", $y, $x;
7. }
8. ?>
```

- a) no output
- b) true 23
- c) true 22
- d) true 33

Answer: b

Explanation: x is preincremented and y is post incremented thus both are 2 in the if condition, later y is increment.

5. What will be the output of the following PHP code?

```
1. <?php
2. $y = 2;
3. $w = 4;
4. $y *= $w /= $y;
5. echo $y, $w;
6. ?>
```

- a) 80.5
- b) 44
- c) 82
- d) 42

Answer: d

Explanation: Expression is evaluated from right to left.

6. What will be the output of the following PHP code?

```

1. <?php
2. $y = 2;
3. if ($y-- == ++$y)
4. {
5.     echo $y;
6. }
7. ?>

```

- a) 2
- b) 1
- c) 3
- d) no output

Answer: a

Explanation: First \$y = 2 is compared to and then decremented, then incremented and compared to \$y = 2.

7. What will be the output of the following PHP code?

```

1. <?php
2. $y = 2;
3. if (**$y == 4)
4. {
5.     echo $y;
6. }
7. ?>

```

- a) 4
- b) 2
- c) error at line2
- d) no output

Answer: c

*Explanation: The ** is not a valid operator, only ++ and — exist.*

8. What will be the output of the following PHP code?

```

1. <?php
2. $y = 2;
3. if (--$y == 2 || $y xor --$y)
4. {
5.     echo $y;
6. }
7. ?>

```

- a) 1
- b) 0

- c) 2
- d) no output

Answer: b

Explanation: $\neg y == 2$ is false but y is decremented, the xor gives true if only one of the operands are true, thus 1 xor 0 is true.

9. What will be the output of the following PHP code?

```
1. <?php
2. $y = 2;
3. if (--$y <> ($y != $y++))
4. {
5.     echo $y;
6. }
7. ?>
```

- a) 1
- b) 0
- c) 2
- d) no output

Answer: b

Explanation: $\neg y == 2$ is false but y is decremented, the xor gives true if only one of the operands are true, thus 1 xor 0 is true.

10. What will be the output of the following PHP code?

```
1. <?php
2. echo $x-- != ++$x;
3. ?>
```

- a) 1
- b) 0
- c) error
- d) no output

Answer: a

Explanation: Automatically x is declared and initialized to 0, then decremented and compared with its increments, thus returns 1.

11. What will be the output of the following PHP code?

```
1. <?php
2. $auth = 1;
3. $status = 1;
4. if ($result = (($auth == 1) && ($status != 0)))
5. {
6.     print "result is $result<br />";
```

7. }

8. ?>

- a) result is true
- b) result is 1
- c) error
- d) no output

Answer: b

Explanation: Result is $x \&\& y$ which returns 1 if both x and y are true.

1. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. while ($i = 10)
4. {
5.     print "hi";
6. }
7. print "hello";
8. ?>
```

- a) hello
- b) infinite loop
- c) hihello
- d) error

Answer: b

Explanation: While condition always gives 1.

2. What will be the output of the following PHP code?

```
1. <?php
2. $i = "";
3. while ($i = 10)
4. {
5.     print "hi";
6. }
7. print "hello";
8. ?>
```

- a) hello
- b) infinite loop
- c) hihello
- d) error

Answer: b

Explanation: While condition always gives 1.

3. What will be the output of the following PHP code?

```
1. <?php
2. $i = 5;
3. while (--$i > 0)
4. {
5.     $i++;
6.     print $i;
7.     print "hello";
8. }
9. ?>
```

- a) 4hello4hello4hello4hello4hello.....infinite
- b) 5hello5hello5hello5hello5hello.....infinite
- c) no output
- d) error

Answer: a

Explanation: i is decremented in the first while execution and then continuously incremented back.

4. What will be the output of the following PHP code?

```
1. <?php
2. $i = 5;
3. while (--$i > 0 && ++$i)
4. {
5.     print $i;
6. }
7. ?>
```

- a) 5
- b) 555555555...infinitely
- c) 54321
- d) error

Answer: b

Explanation: As it is && operator it is being incremented and decremented continuously.

5. What will be the output of the following PHP code?

```
1. <?php
2. $i = 5;
3. while (--$i > 0 || ++$i)
4. {
5.     print $i;
```



```
6. }  
7. ?>
```

- a) 5432111111....infinitely
- b) 555555555...infinitely
- c) 54321
- d) 5

Answer: a

Explanation: As it is || operator the second expression is not evaluated till i becomes 1 then it goes into a loop.

6. What will be the output of the following PHP code?

```
1. <?php  
2. $i = 0;  
3. while(++$i || --$i)  
4. {  
5.     print $i;  
6. }  
7. ?>
```

- a) 1234567891011121314....infinitely
- b) 01234567891011121314...infinitely
- c) 1
- d) 0

Answer: a

Explanation: As it is || operator the second expression is not evaluated and i is always incremented, in the first case to 1.

7. What will be the output of the following PHP code?

```
1. <?php  
2. $i = 0;  
3. while (++$i && --$i)  
4. {  
5.     print $i;  
6. }  
7. ?>
```

- a) 1234567891011121314....infinitely
- b) 01234567891011121314...infinitely
- c) no output
- d) error

Answer: c

Explanation: The first condition itself fails thus the loop exits.

8. What will be the output of the following PHP code?

```
1. <?php
```

```

2. $i = 0;

3. while ((--$i > ++$i) - 1)

4. {

5.     print $i;

6. }

7. ?>

```

- a) 00000000000000000000....infinitely
- b) -1-1-1-1-1-1-1-1-1-1...infinitely
- c) no output
- d) error

Answer: a

Explanation: ($-\$i > ++\i) evaluates to 0 but -1 makes it enters the loop and prints i which is 0.

9. What will be the output of the following PHP code?

```

1. <?php

2. $i = 2;

3. while (++$i)

4. {

5.     while ($i --> 0)

6.         print $i;

7. }

8. ?>

```

- a) 210
- b) 10
- c) no output
- d) infinite loop

Answer: a

Explanation: The loop ends when i becomes 0.

10. What will be the output of the following PHP code?

```

1. <?php

2. $i = 2;

3. while (++$i)

4. {

5.     while (--$i > 0)

6.         print $i;

7. }

8. ?>

```

- a) 210
- b) 10
- c) no output
- d) infinite loop

Answer: d

Explanation: The loop never ends as *i* is always incremented and then decremented.

1. What will be the output of the following PHP code?

```
1. <?php
2. echo 5 * 9 / 3 + 9;
3. ?>
```

- a) 24
- b) 3.7
- c) 3.85
- d) 0

Answer: a

Explanation: Operator precedence order must be followed.

2. What will be the output of the following PHP code?

```
1. <?php
2. echo 5 * 9 / 3 + 9
3. ?>
```

- a) 24
- b) 3.7
- c) 3.85
- d) 0

Answer: a

Explanation: Operator precedence order must be followed.

3. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. $j = 0;
4. if ($i && ($j = $i + 10)) {
5.     echo "true";
6. }
7. echo $j;
8. ?>
```

- a) 10
- b) 0
- c) true0
- d) true10

Answer: b

Explanation: In if condition when the first case is 0 and is an && operation then the second command is not executed.

4. What will be the output of the following PHP code?

```
1. <?php
2. $i = 10;
3. $j = 0;
4. if ($i || ($j = $i + 10)) {
5.     echo "true";
6. }
7. echo $j;
8. ?>
```

- a) 20
- b) true0
- c) 0
- d) true20

Answer: b

Explanation: In if condition when the first case is 1 and is an || operation then the second command is not executed.

5. What will be the output of the following PHP code?

```
1. <?php
2. $i = 1;
3. if ($i++ && ($i == 1))
4.     printf("Yes\n$i");
5. else
6.     printf("No\n$i");
7. ?>
```

- a) No 2
- b) Yes 1
- c) Yes 2
- d) No 1

Answer: a

Explanation: The first condition returns true and increments but the second condition is false.

6. What will be the output of the following PHP code?

```
1. <?php
2. $a = 1; $b = 3;
3. $d = $a++ + ++$b;
4. echo $d;
5. ?>
```

- a) 5
- b) 4
- c) 3
- d) error

Answer: a

Explanation: Post increment of a is done after expression evaluation.

7. What will be the output of the following PHP code?

```
1. <?php
2. $a = 1; $b = 1; $d = 1;
3. print ++$a + ++$a+$a++; print $a++ + ++$b; print ++$d + $d++ + $a++;
4. ?>
```

- a) 869
- b) 742
- c) 368
- d) error

Answer: a

Explanation: Follow the order of post and pre increments.

8. What will be the output of the following PHP code?

```
1. <?php
2. $a = 10; $b = 10;
3. if ($a = 5)
4.     $b--;
5. print $a;print $b--;
6. ?>
```

- a) 58
- b) 59
- c) 109
- d) 108

Answer: b

Explanation: a is set to 5 in the if condition and b is post decremented in the print statement.

9. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. $x = $i++; $y = ++$i;
4. print $x; print $y;
5. ?>
```

- a) 02
- b) 12

- c) 01
- d) 21

Answer: a

Explanation: First case i is incremented after setting x to i.

10. What will be the output of the following PHP code?

```
1. <?php
2.  $a = 5; $b = -7; $c =0;
3.  $d = ++$a && ++$b || ++$c;
4.  print $d; print $a;
5. ?>
```

- a) 16
- b) 06
- c) 15
- d) 05

Answer: a

Explanation: 1&&0||1 is evaluated to 1 and the a is also preincremented to 6.

11. What will be the output of the following PHP code?

```
1. <?php
2.  $b = 1; $c = 4; $a = 5;
3.  $d = $b + $c == $a;
4.  print $d;
5. ?>
```

- a) 5
- b) 0
- c) 10
- d) 1

Answer: d

Explanation: First b and c are added and then tested if d=5, which is true thus return 1.

1. What will be the output of the following PHP code?

```
1. <?php
2.  echo 5 * 9 / 3 + 9;
3. ?>
```

- a) 24
- b) 3.7
- c) 3.85
- d) 0

Answer: a

Explanation: Operator precedence order must be followed.

2. What will be the output of the following PHP code?

```
1. <?php
2. $var1 = 1 + ++5;
3. echo $var1;
4. ?>
```

- a) no output
- b) error
- c) 6
- d) 7

Answer: b

Explanation: Operator ++ can be done only on variables.

3. What will be the output of the following PHP code?

```
1. <?php
2. $var1 = 0;
3. $var1 = ($var1 + 5)++;
4. echo $var1;
5. ?>
```

- a) 5
- b) error
- c) 6
- d) 7

Answer: b

Explanation: Operator ++ can be done only on variables.

4. What will be the output of the following PHP code?

```
1. <?php
2. $var1 = 0;
3. $var1 = $var1++ + 5;
4. echo $var1;
5. ?>
```

- a) 5
- b) error
- c) 6
- d) 7

Answer: a

Explanation: Operator precedence followed.

5. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $var1 = 0;  
3. $var1 = ++$var1 + 5;  
4. echo $var1;  
5. ?>
```

- a) 5
- b) error
- c) 6
- d) 7

Answer: c

Explanation: Operator precedence followed.

6. What will be the output of the following PHP code?

```
1. <?php  
2. $var1 = 0;  
3. $var1 = $var1 + 5;  
4. echo $var1++;  
5. ?>
```

- a) 5
- b) error
- c) 6
- d) 7

Answer: a

Explanation: Operator precedence followed, incremented after display.

7. What will be the output of the following PHP code?

```
1. <?php  
2. $var1 = 1;  
3. echo $var1 = ++$var1 % 2 + ++$var1;  
4. ?>
```

- a) 1
- b) 0
- c) 2
- d) 3

Answer: d

Explanation: Evaluation done from right to left.

8. What will be the output of the following PHP code?

```
1. <?php  
2. $a = 5;$b = -7;$c =0;  
3. $d = ++$a && ++$b || ++$c;  
4. print $d;print $a;
```


5. ?>

- a) 16
- b) 06
- c) 15
- d) 05

Answer: a

Explanation: $1 \&\&0 || 1$ is evaluated to 1 and the a is also pre incremented to 6.

9. What will be the output of the following PHP code?

```
1. <?php
2. $var1 = 3;
3. print $var = ++$var;
4. ?>
```

- a) 1
- b) 0
- c) 2
- d) 3

Answer: a

Explanation: $\$var = ++\var returns 1(success).

10. What will be the output of the following PHP code?

```
1. <?php
2. $var1 = 3;
3. print ++$var++;
4. ?>
```

- a) 3
- b) 4
- c) 5
- d) error

Answer: d

Explanation: First pre increment is done and the result is a number, thus post increment cannot be performed on it.

1. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0; $j = 1; $k = 2;
3. print !((($i + $k) < ($j - $k)));
4. ?>
```

- a) 1
- b) true
- c) false
- d) 0

Answer: a

Explanation: True is 1.

2. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;$j = 1;$k = 2;
3. print !(( + + $i + $j) > ($j - $k));
4. ?>
```

- a) 1
- b) no output
- c) error
- d) 0

Answer: b

Explanation: The equation outputs false .

3. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;$j = 1;$k = 2;
3. print (( + + $i + $j) >! ($j - $k));
4. ?>
```

- a) 1
- b) no output
- c) error
- d) 0

Answer: a

Explanation: Negation of a number is 0.

4. What will be the output of the following PHP code?

```
1. <?php
2. $a = 0x6db7;
3. print $a<<6;
4. ?>
```

- a) 1797568
- b) no output
- c) error
- d) 0x6dc0

Answer: a

Explanation: The output is in decimal.

5. What will be the output of the following PHP code?

```
1. <?php
2. $a = 'a' ;
```

```
3. print $a * 2;
```

```
4. ?>
```

- a) 192
- b) 2
- c) error
- d) 0

Answer: d

Explanation: Characters cannot be multiplied.

6. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = '4' ;
```

```
3. print + + $a;
```

```
4. ?>
```

- a) no output
- b) error
- c) 5
- d) 0

Answer: c

Explanation: The character is type casted to integer before multiplying.

7. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = '12345';
```

```
3. print "qwe{$a}rty";
```

```
4. ?>
```

- a) qwe12345rty
- b) qwe{\$a}rty
- c) error
- d) no output

Answer: a

Explanation: {\$}dereferences the variable within.

8. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = '12345';
```

```
3. print "qwe".$a."rty";
```

```
4. ?>
```

- a) qwe12345rty
- b) qwe\$rty
- c) error
- d) no output

Answer: a

Explanation: . dereferences the variable/string within.

9. What will be the output of the following PHP code?

```
1. <?php
2. $a  =  '12345';
3. echo 'qwe{$a}rty';
4. ?>
```

- a) qwe12345rty**
- b) qwe{\$a}rty**
- c) error**
- d) no output**

Answer: b

Explanation: qwe{\$a}rty, single quotes are not parsed.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a  =  '12345';
3. echo "qwe$aarty";
4. ?>
```

- a) qwe12345rty**
- b) qwe\$aarty**
- c) qwe**
- d) error**

Answer: c

Explanation: qwe, because \$a became \$arty, which is undefined.

1. What will be the output of the following PHP code?

```
1. <?php
2. $x;
3. if ($x)
4.     print "hi" ;
5. else
6.     print "how are u";
7. ?>
```

- a) how are u**
- b) hi**
- c) error**
- d) no output**

Answer: a

Explanation: Uninitialized x is set to 0, thus if condition fails.

2. What will be the output of the following PHP code?

```
1. <?php
2. $x = 0;
3. if ($x++)
4.     print "hi";
5. else
6.     print "how are u";
7. ?>
```

- a) hi
- b) no output
- c) error
- d) how are u

Answer: d

Explanation: x is incremented after if which evaluates to false.

3. What will be the output of the following PHP code?

```
1. <?php
2. $x;
3. if ($x == 0)
4.     print "hi" ;
5. else
6.     print "how are u";
7.     print "hello"
8. ?>
```

- a) how are uhello
- b) hihello
- c) hi
- d) no output

Answer: b

Explanation: else condition without brackets performs the following statements only.

4. What will be the output of the following PHP code?

```
1. <?php
2. $x = 0;
3. if ($x == 1)
4.     if ($x >= 0)
5.         print "true";
6.     else
```

```
7.         print "false";
8. ?>
```

- a) true
- b) false
- c) error
- d) no output

Answer: d

Explanation: The nested for loop is not entered if outer condition is false.

5. What will be the output of the following PHP code?

```
1. <?php
2. $a = 1;
3. if ($a--)
4.     print "True";
5. if ($a++)
6.     print "False";
7. ?>
```

- a) true
- b) false
- c) error
- d) no output

Answer: a

Explanation: Due to post increment and post decrement only the first condition is satisfied.

6. What will be the output of the following PHP code?

```
1. <?php
2. $a = 1;
3. if (echo $a)
4.     print "True";
5. else
6.     print "False";
7. ?>
```

- a) true
- b) false
- c) error
- d) no output

Answer: c

Explanation: echo does not return anything so if condition is empty.

7. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = 1;
3. if (print $a)
4.     print "True";
5. else
6.     print "False";
7. ?>
```

- a) true
- b) false
- c) error
- d) no output

Answer: a

Explanation: print returns 1 if it prints anything.

8. What will be the output of the following PHP code?

```
1. <?php
2. $a = 10;
3. if (1)
4.     print "all";
5. else
6.     print "some"
7. else
8.     print "none";
9. ?>
```

- a) all
- b) some
- c) error
- d) none

Answer: c

Explanation: Hanging else statement.

9. What will be the output of the following PHP code?

```
1. <?php
2. $a = 10;
3. if (0)
4.     print "all";
5. if
6.     else
7.     print "some"
```

8. ?>

- a) all
- b) some
- c) error
- d) no output

Answer: c

Explanation: No else statement to end the if statement.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a = "";
3. if ($a)
4.     print "all";
5. if
6. else
7.     print "some";
8. ?>
```

- a) all
- b) some
- c) error
- d) no output

Answer: b

Explanation: Empty string is evaluated to 0.

11. What will be the output of the following PHP code?

```
1. <?php
2. $a = "a";
3. if ($a)
4.     print "all";
5. else
6.     print "some";
7. ?>
```

- a) all
- b) some
- c) error
- d) no output

Answer: a

Explanation: The value of a is evaluated to 1 as it has a value.

1. What will be the output of the following PHP code?

```
1. <?php
```



```
2. $x = 10;
3. $y = 20;
4. if ($x > $y + $y != 3)
5.     print "hi" ;
6. else
7.     print "how are u";
8. ?>
```

- a) how are u
- b) hi
- c) error
- d) no output

Answer: b

Explanation: Expression evaluates to true.

2. What will be the output of the following PHP code?

```
1. <?php
2. $x = 10;
3. $y = 20;
4. if ($x > $y && 1||1)
5.     print "hi" ;
6. else
7.     print "how are u";
8. ?>
```

- a) how are u
- b) hi
- c) error
- d) no output

Answer: b

Explanation: Expression evaluates to true.

3. What will be the output of the following PHP code?

```
1. <?php
2. $x = 10;
3. $y = 20;
4. if ($x > $y && 1||1)
5.     print "hi" ;
6. else
7.     print "how are u";
```

8. ?>

- a) how are u
- b) hi
- c) error
- d) no output

Answer: b

Explanation: Expression evaluates to true.

4. What will be the output of the following PHP code?

```
1. <?php
2. if (-100)
3.     print "hi" ;
4. else
5.     print "how are u";
6. ?>
```

- a) how are u
- b) hi
- c) error
- d) no output

Answer: b

Explanation: Expression evaluates to true.

5. What will be the output of the following PHP code?

```
1. <?php
2. if (0.1)
3.     print "hi" ;
4. else
5.     print "how are u";
6. ?>
```

- a) how are u
- b) hi
- c) error
- d) no output

Answer: b

Explanation: Expression evaluates to true.

6. What will be the output of the following PHP code?

```
1. <?php
2. if (0.0)
3.     print "hi" ;
4. else
```

```
5.     print "how are u";
```

```
6. ?>
```

- a) how are u
- b) hi
- c) error
- d) no output

Answer: a

Explanation: Expression evaluates to false.

7. What will be the output of the following PHP code?

```
1. <?php
```

```
2. if (print "0")
```

```
3.     print "hi" ;
```

```
4. else
```

```
5.     print "how are u";
```

```
6. ?>
```

- a) 0how are u
- b) 0hi
- c) hi
- d) how are u

Answer: b

Explanation: Expression evaluates to true as print returns 1.

8. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $x = 1;
```

```
3. if ($x == 2)
```

```
4.     print "hi" ;
```

```
5. else if($x = 2)
```

```
6.     print $x;
```

```
7. else
```

```
8.     print "how are u";
```

```
9. ?>
```

- a) error
- b) 2
- c) hi
- d) how are u

Answer: b

Explanation: Enters if else as first condition is false and thus x is set to 2.

9. What will be the output of the following PHP code?

```
1. <?php
2. $x = 1;
3. if ($x = $x&0)
4.     print $x ;
5. else
6.     print "how are u";
7. ?>
```

- a) 0
- b) 1
- c) error
- d) how are u

Answer: d

Explanation: $x \& 0$ is 0, thus evaluated to false.

10. What will be the output of the following PHP code?

```
1. <?php
2. $x = 1;
3. if ($x = $x&0)
4.     print $x ;
5. else
6.     print "how are u";
7. ?>
```

- a) 0
- b) 1
- c) error
- d) how are u

Answer: d

Explanation: $x \& 0$ is 0, thus evaluated to false.

11. What will be the output of the following PHP code?

```
1. <?php
2. $x = 1;
3. if ($x = $x&0)
4.     print $x;
5. else
6.     break;
7. ?>
```

- a) 0
- b) 1

- c) error
- d) no output

Answer: c

Explanation: break is not defined for a if else ladder.

12. What will be the output of the following PHP code?

```
1. <?php
2. $a = 100;
3. if ($a > 10)
4.     printf("M.S. Dhoni");
5. else if ($a > 20)
6.     printf("M.E.K Hussey");
7. else if($a > 30)
8.     printf("A.B. de villiers");
9. ?>
```

Answer: a

Explanation: In if else if one condition is satisfied then no other condition is checked.

1. What will be the output of the following PHP code?

```
1. <?php
2. $x = 10;
3. $y = 5;
4. $z = 3;
5. if ($x / $y / $z)
6.     print "hi";
7. else
8.     print "hello";
9. ?>
```

- a) hi
- b) hello
- c) error
- d) no output

Answer: a

Explanation: In php division returns a float that is a non zero value thus evaluates to true.

2. What will be the output of the following PHP code?

```
1. <?php
2. if (!print "hi")
3.     if (print "hello")
```

```
4. print "hi";
```

```
5. ?>
```

- a) hi
- b) hihellohi
- c) hihi
- d) no output

Answer: a

Explanation: Print returns true and thus the first if statement also is not executed.

3. What will be the output of the following PHP code?

```
1. <?php
```

```
2. if (print "hi" - 1)
```

```
3.     print "hello"
```

```
4. ?>
```

- a) hi
- b) hihello
- c) error
- d) no output

Answer: c

Explanation: print returns true and when 1 is subtracted it is syntax error.

4. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $x = 1;
```

```
3. if ($x--)
```

```
4.     print "hi"
```

```
5.     $x--;
```

```
6. else
```

```
7.     print "hello"
```

```
8. ?>
```

- a) hi
- b) hello
- c) error
- d) no output

Answer: c

Explanation: The if statement has no brackets and it expects a else/else if after a if.

5. What will be the output of the following PHP code?

```
1. <?php
```

```
2. $a = 10;
```

```
3. $b = 11;
```

```
4. if ($a < ++$a || $b < ++$b)
5.     print "hello";
6. else
7.     print "hi";
8. ?>
```

- a) hi
- b) hello
- c) error
- d) no output

Answer: a

Explanation: The operator precedence of ++ is higher than <, thus the increment happens first and then compared.

6. What will be the output of the following PHP code?

```
1. <?php
2. $a = 2;
3. if ($a-- - --$a - $a)
4.     print "hello";
5. else
6.     print "hi";
7. ?>
```

- a) hi
- b) hello
- c) error
- d) no output

Answer: b

Explanation: Computing the expression in the if clause, it sums up to 2 which is a positive value.

7. What will be the output of the following PHP code?

```
1. <?php
2. $a = 2;
3. if ($a-- - --$a - $a)
4.     print "hello";
5. else
6.     print "hi";
7. ?>
```

- a) hi
- b) hello
- c) error
- d) no output

Answer: b

Explanation: Computing the expression in the if clause, it sums upto to 2 which is a positive value.

8. What will be the output of the following PHP code?

```
1. <?php
2. $a = "hello";
3. if ($a.length)
4.     print $a.length;
5. else
6.     print "hi";
7. ?>
```

a) hellolength

b) 5

c) hi

d) error

Answer: a

Explanation: The . operator appends a string and returns true.

9. What will be the output of the following PHP code?

```
1. <?php
2. $a = "hello";
3. if (strlen($a))
4.     print strlen($a);
5. else
6.     print "hi";
7. ?>
```

a) hellolength

b) 5

c) hi

d) error

Answer: b

Explanation: The function strlen(\$a) gives the length of the string, 5, which is considered true.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a = "1";
3. $b = "0";
4. if ((int)$a && $b)
5.     print"hi";
6. else
```



```
7.      print "hello";
```

```
8. ?>
```

- a) hello
- b) no output
- c) hi
- d) error

Answer: a

Explanation: The expression is evaluated with values contained in the string, even without typecasting.

1. What will be the output of the following PHP code?

```
1. <?php
2. $a = "1";
3. switch ($a)
4. {
5. case 1:
6.     print "hi";
7. case 2:
8.     print "hello";
9. default:
10.    print "hi1";
11. }
12. ?>
```

- a) hihellohi1
- b) hi
- c) hihi1
- d) hi1

Answer: a

Explanation: As break is not provided it executes all the cases.

2. What will be the output of the following PHP code?

```
1. <?php
2. $a = "2";
3. switch ($a)
4. {
5. case 1:
6.     print "hi";
7. case 2:
8.     print "hello";
```

```
9.         break;

10. default:

11.         print "hi1";

12. }

13. ?>
```

- a) hihellohi1
- b) hello
- c) hihi1
- d) hi1

Answer: b

Explanation: As hello is provided after case2 it breaks the loop.

3. What will be the output of the following PHP code?

```
1. <?php
2. $a = "1";
3. switch($a)
4. {
5. case 1:
6.     break;
7.     print "hi";
8. case 2:
9.     print "hello";
10.    break;
11. default:
12.    print "hi1";
13. }
14. ?>
```

- a) hihellohi1
- b) no output
- c) hihi1
- d) hi1

Answer: b

Explanation: As break is provided before print statement in case 2 it breaks the loop before printing.

4. What will be the output of the following PHP code?

```
1. <?php
2. $a = "1";
3. $a = 1;
```

```
4. $b = 1;
5. switch($a)
6. {
7. case $a * $b:
8.     print "hi";
9.     break;
10. case $a / $b:
11.     print "hello";
12.     break;
13. default:
14.     print "hi1";
15. }
16. ?>
```

- a) hihellohi1
- b) hi
- c) hihello
- d) hi1

Answer: b

Explanation: It checks the first case, when it finds it equal it will perform it breaks out.

5. What will be the output of the following PHP code?

```
1. <?php
2. $a = 97;
3. switch($a)
4. {
5. case "a":
6.     print "hi";
7.     break;
8. case 97:
9.     print "hello";
10.    break;
11. default:
12.    print "hi1";
13. }
14. ?>
```

- a) hihellohi1

- b) hi
- c) hihello
- d) hello

Answer: d

Explanation: Downcasting does not happen in case, it compares only with its data type.

6. What will be the output of the following PHP code?

```
1. <?php
2. $b = 1;
3. switch($b)
4. {
5.     case 1.0:
6.         print "hi";
7.         break;
8.     case 1:
9.         print "hello";
10.        break;
11.    default:
12.        print "hi1";
13. }
14. ?>
```

- a) hihellohi1
- b) hi
- c) hihello
- d) hello

Answer: a

Explanation: Upcasting does happen in case, it compares it with 1.0 and thus prints hi and exits.

7. What will be the output of the following PHP code?

```
1. <?php
2. const $b = 1;
3. switch($b)
4. {
5.     case 1:
6.         print "hi";
7.         break;
8.     case 1:
9.         print "hello";
```

```
10.         break;

11. default:

12.         print "hi1";

13. }

14. ?>
```

- a) error
- b) hi
- c) hihello
- d) hello

Answer: a

Explanation: Constants cannot be used in switch cases.

8. What will be the output of the following PHP code?

```
1. <?php
2. $b = 1;
3. switch(print $b)
4. {
5. case 2:
6.     print "hello";
7.     break;
8. case 1:
9.     print "hi";
10.    break;
11. default:
12.    print "hi1";
13. }
14. ?>
```

- a) 1hello
- b) 1hi
- c) 1hi1
- d) error

Answer: b

Explanation: Print returns 1, thus it gives case 1.

9. What will be the output of the following PHP code?

```
1. <?php
2. switch($b)
3. {
```

```
4. case 2:
5.     print "hello";
6.     break;
7. case 1:
8.     print "hi";
9.     break;
10. }
11. ?>
```

- a) hello
- b) hi
- c) no output
- d) error

Answer: c

Explanation: If that case does not exist then it searches for default and on not finding it does nothing.

10. What will be the output of the following PHP code?

```
1. <?php
2. switch($b)
3. {
4. case 2:
5.     print "hello";
6.     break;
7. case b:
8.     print "hi";
9.     break;
10. }
11. ?>
```

- a) hello
- b) hi
- c) no output
- d) error

Answer: c

Explanation: Case cannot be defined by a variable.

1. What will be the output of the following PHP code?

```
1. <?php
2. while()
3. {
```

```
4.     print "hi";
5. }
6. ?>
```

- a) infinite loop
- b) hi
- c) no output
- d) error

Answer: d

Explanation: The while loop cannot be defined without a condition.

2. What will be the output of the following PHP code?

```
1. <?php
2. do
3. {
4.     print "hi";
5. }
6. while(0);
7. print "hello";
8. ?>
```

- a) infinite loop
- b) hihello
- c) hello
- d) error

Answer: b

Explanation: The do while loop executes at least once as the condition is in the while loop.

3. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. do
4. {
5.     print "hi";
6.     $i++;
7. }
8. while ($i != 3);
9. ?>
```

Answer: c

Explanation: The check happens after the increment, thus it prints until i = 4.

4. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. while ($i != 3)
4. {
5.     print "hi";
6.     $i++;
7. }
8. ?>
```

Answer: b

Explanation: The check happens before the increment, thus it prints until $i = 3$.

5. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. while ($i < 3)
4. {
5.     print "hi";
6.     $i--;
7. }
8. print "hello"
9. ?>
```

Answer: d

Explanation: There is no increment of i making it infinite.

6. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. while ($i < 3)
4. {
5.     $i++;
6. }
7. print $i;
8. ?>
```

- a) 2
- b) 3
- c) 0
- d) 1

Answer: b

Explanation: The increment happens and then the check happens.

7. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. do
4. {
5.     $i++;
6. }
7. while ($i < 3);
8. print $i;
9. ?>
```

a) 2

b) 3

c) 0

d) 1

Answer: b

Explanation: The increment happens and then the check happens.

8. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0
3. while ($i++)
4. {
5.     print $i;
6. }
7. print $i;
8. ?>
```

a) 0

b) infinite loop

c) 01

d) 1

Answer: d

Explanation: As it is a post increment, it checks and then does not enter the loop, thus prints only 1.

9. What will be the output of the following PHP code?

```
1. <?php
2. $i = "";
3. while($i)
```

```
4. {  
5.     print "hi";  
6. }  
  
7. print "hello";  
8. ?>
```

- a) hello
- b) infinite loop
- c) hihello
- d) error

Answer: a

Explanation: While accept does not accept anything other than a 0 or any other number as false and true.

10. What will be the output of the following PHP code?

```
1. <?php  
2. $i = "";  
3. while ($i)  
4. {  
5.     print "hi";  
6. }  
  
7. while($i < 8)  
8.     $i++;  
9. print "hello";  
10. ?>
```

- a) Hi is printed 8 times, hello 7 times and then hi 2 times
- b) Hi is printed 10 times, hello 7 times
- c) Hi is printed once, hello 7 times
- d) Hi is printed once, hello 7 times and then hi 2 times

Answer: d

Explanation: The while loop ends only when a } is encountered.

1. What will be the output of the following PHP code?

```
1. <?php  
2. $i = 0;  
3. while($i = 10)  
4. {  
5.     print "hi";  
6. }  
  
7. print "hello";
```

8. ?>

- a) hello
- b) infinite loop
- c) hihello
- d) error

Answer: b

Explanation: While condition always gives 1.

2. What will be the output of the following PHP code?

```
1. <?php
2. $i = "";
3. while ($i = 10)
4. {
5.     print "hi";
6. }
7. print "hello";
8. ?>
```

- a) hello
- b) infinite loop
- c) hihello
- d) error

Answer: b

Explanation: While condition always gives 1.

3. What will be the output of the following PHP code?

```
1. <?php
2. $i = 5;
3. while (--$i > 0)
4. {
5.     $i++; print $i; print "hello";
6. }
7. ?>
```

- a) 4hello4hello4hello4hello4hello.....infinite
- b) 5hello5hello5hello5hello5hello.....infinite
- c) no output
- d) error

Answer: b

Explanation: i is decremented in the while loop in the condition check and then incremented back.

4. What will be the output of the following PHP code?

```
1. <?php
```

```

2. $i = 5;

3. while (--$i > 0 && ++$i)

4. {

5.     print $i;

6. }

7. ?>

```

- a) 5
- b) 555555555...infinitely
- c) 54321
- d) error

Answer: b

Explanation: As it is && operator it is being incremented and decremented continuously.

5. What will be the output of the following PHP code?

```

1. <?php

2. $i = 5;

3. while (--$i > 0 || ++$i)

4. {

5.     print $i;

6. }

7. ?>

```

- a) 5432111111....infinitely
- b) 555555555...infinitely
- c) 54321
- d) 5

Answer: a

Explanation: As it is || operator the second expression is not evaluated till i becomes 1 then it goes into a loop.

6. What will be the output of the following PHP code?

```

1. <?php

2. $i = 0;

3. while(++$i || --$i)

4. {

5.     print $i;

6. }

7. ?>

```

- a) 1234567891011121314....infinitely
- b) 01234567891011121314...infinitely
- c) 1

d) 0

Answer: a

Explanation: As it is || operator the second expression is not evaluated and i is always incremented, in the first case to 1.

7. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. while (++$i && --$i)
4. {
5.     print $i;
6. }
7. ?>
```

- a) 1234567891011121314....infinitely**
- b) 01234567891011121314...infinitely**
- c) no output**
- d) error**

Answer: c

Explanation: The first condition itself fails thus the loop exists.

8. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. while ((--$i > ++$i) - 1)
4. {
5.     print $i;
6. }
7. ?>
```

- a) 00000000000000000000....infinitely**
- b) -1-1-1-1-1-1-1-1-1-1...infinitely**
- c) no output**
- d) error**

Answer: a

Explanation: (-\$i > ++\$i) evaluates to 0 but -1 makes it enters the loop and prints i.

9. What will be the output of the following PHP code?

```
1. <?php
2. $i = 2;
3. while (++$i)
4. {
```

```
5.         while ($i --> 0)
6.             print $i;
7. }
8. ?>
```

- a) 210
- b) 10
- c) no output
- d) infinite loop

Answer: a

Explanation: The loop ends when i becomes 0.

10. What will be the output of the following PHP code?

```
1. <?php
2. $i = 2;
3. while (++$i)
4. {
5.     while (--$i > 0)
6.         print $i;
7. }
```

- a) 210
- b) 10
- c) no output
- d) infinite loop

Answer: d

Explanation: The loop never ends as i is always incremented and then decremented.

1. What will be the output of the following PHP code?

```
1. <?php
2. $i = 0;
3. for ($i)
4. {
5.     print $i;
6. }
```

- a) 0
- b) infinite loop
- c) no output
- d) error

Answer: d

Explanation: Wrong syntax for for loop.

2. What will be the output of the following PHP code?

```
1. <?php
2. $colors = array("red","green","blue","yellow");
3. foreach ($colors as $value)
4. {
5.     echo "$value <br>";
6. }
7. ?>
```

Answer: a

Explanation: This runs a for loop for that array.

3. What will be the output of the following PHP code?

```
1. <?php
2. for ($x = 0; $x <= 10; $x++)
3. {
4.     echo "The number is: $x <br>";
5. }
6. ?>
```

Answer: a

Explanation: This runs a for loop from 0 to 10.

4. What will be the output of the following PHP code?

```
1. <?php
2. for ($x = 0; $x <= 10; print ++$x)
3. {
4.     print ++$x;
5. }
6. ?>
```

a) 123456789101112

b) 12345678910

c) 1234567891011

d) infinite loop

Answer: a

Explanation: The value of x is incremented and printed twice before checking, this last loop it prints 11 and 12.

5. What will be the output of the following PHP code?

```
1. <?php
```

```
2. for ($x = 1; $x < 10; ++$x)
3. {
4.     print "*\t";
5. }
6. ?>
```

- a) *****
- b) *****
- c) *****
- d) infinite loop

Answer: b

Explanation: Loop runs from 1 to 9 i.e 9 times.

6. What will be the output of the following PHP code?

```
1. <?php
2. for ($x = -1; $x < 10; --$x)
3. {
4.     print $x;
5. }
6. ?>
```

- a) 123456789101112
- b) 12345678910
- c) 1234567891011
- d) infinite loop

Answer: d

Explanation: The value of x is decremented thus making it an infinite loop.

7. What will be the output of the following PHP code?

```
1. <?php
2. $x;
3. for ($x = -3; $x < -5; ++$x)
4. {
5.     print ++$x;
6. }
7. ?>
```

- a) -3-4-5
- b) -3-4
- c) infinite loop
- d) no output

Answer: d

Explanation: The loop is not even entered as x is initially 0.

8. What will be the output of the following PHP code?

```
1. <?php
2. for ($i++; $i == 1; $i = 2)
3.     print "In for loop ";
4. print "After loop\n";
5.
6. ?>
```

- a) In for loop
- b) After for loop
- c) In for loopAfter for loop
- d) Infinite loop

Answer: c

Explanation: The loop runs only once as value of x is incremented.

9. What will be the output of the following PHP code?

```
1. <?php
2. for (1; $i == 1; $i = 2)
3.     print "In for loop ";
4. print "After loop\n";
5. ?>
```

- a) In for loop
- b) After for loop
- c) In for loopAfter for loop
- d) Infinite loop

Answer: b

Explanation: The loop does not run as i initialized in check statement will be zero.

10. What will be the output of the following PHP code?

```
1. <?php
2. for ($i == 2; ++$i == $i; ++$i)
3.     print "In for loop ";
4. print "After loop\n";
5. ?>
```

- a) In for loopIn for loopIn for loopIn for loop.....infinitely
- b) After for loopAfter for loopAfter for loop.....infinitely
- c) In for loopAfter for loopIn for loopAfter for loopIn for loopAfter for loop.....infinitely
- d) After for loop

Answer: a

Explanation: The loop never exits as the condition $++X == X$ is always satisfied,evaluated from right to left.

1. What will be the output of the following PHP code?

```
1. <?php
2. for ($x = 1; $x < 10; $x++)
3.     for ($y = 1; $y < 5; $y++)
4.         print "Hello";
5. ?>
```

- a) Hello....36 times
- b) Hello....45 times
- c) Hello....50 times
- d) Hello....40 times

Answer: a

*Explanation: 9*4 times is printed.*

2. What will be the output of the following PHP code?

```
1. <?php
2. for ($count = 1; $count != 20;$count++)
3. {
4.     print $count;
5.     $count++;
6. }
7. ?>
```

- a) Infinite
- b) 123.....20
- c) 1357...19
- d) 13579...21

Answer: a

Explanation: Condition always fails as count takes only odd numbers.

3. What will be the output of the following PHP code?

```
1. <?php
2. for ($count = 1; $count < 20; $count++);
3.     print $count;
4. ?>
```

- a) 20
- b) 19
- c) 12345678910....19
- d) 12345678910....1920

Answer: a

Explanation: The for loop has no body, it just runs till condition is satisfied.

4. What will be the output of the following PHP code?

```
1. <?php
```

```
2. for ($count = 0; $count < 3;$count++);  
3. {  
4.     print "hi";continue;print "hello";  
5. }  
6. ?>
```

- a) hihhi
- b) hihellohihellohihello
- c) hellohellohello
- d) hi

Answer: a

Explanation: When continue is encountered it skips to the next iteration.

5. What will be the output of the following PHP code?

```
1. <?php  
2. for ($count = 0; $count<3;$count++);  
3. {  
4.     print "hi";break;print "hello";  
5. }  
6. ?>
```

- a) hihhi
- b) hihellohihellohihello
- c) hellohellohello
- d) hi

Answer: d

Explanation: When break is encountered it leaves the loop.

6. What will be the output of the following PHP code?

```
1. <?php  
2. for(++$i; ++$i; ++$i)  
3. {  
4.     print $i;  
5.     if ($i == 4)  
6.         break;  
7. }  
8. ?>
```

- a) 24
- b) 134
- c) 1234
- d) 1

Answer: a

Explanation: The order of execution is initialization, check, increment/decrement, check, increment/decrement, check, increment/decrement.....so on.

7. What will be the output of the following PHP code?

```
1. <?php
2. for ($i = 0;$i = -1;$i = 1)
3. {
4.     print $i;
5.     if ($i != 1)
6.         break;
7. }
8. ?>
```

- a) 0
- b) infinite loop**
- c) -1
- d) 1

Answer: c

Explanation: The order of execution is initialization, check, increment/decrement, check, increment/decrement, check, increment/decrement.....so on.

8. What will be the output of the following PHP code?

```
1. <?php
2. for(;;)
3. {
4.     print "10";
5. }
6. ?>
```

- a) 10**
- b) infinite loop
- c) no output
- d) error

Answer: b

Explanation: There is no check condition to stop the execution of the loop.

9. What will be the output of the following PHP code?

```
1. <?php
2. for ($i = 0; $i < 3; $i++)
3. {
4.     for($j = $i; $j > 0; $j--)
```

```

5.         print " ";
6.     for($k = $j; $k < 3; $k++)
7.         print "*";
8.         print "\n";
9. }
10. ?>

```

Answer: a

Explanation: Follow the trace of i, j prints 3 – i no of spaces for each i, k prints i stars for each loop.

10. What will be the output of the following PHP code?

```

1. <?php
2. for ($i = 0; -5 ; $i++)
3. {
4.     print "i";
5.     if ($i == 3)
6.         break;
7. }
8. ?>

```

- a) 0 1 2 3 4
- b) 0 1 2 3
- c) 0 1 2 3 4 5
- d) error

Answer: b

Explanation: The break statement after breaks the loop after i=3, does not print anymore.

1. What will be the output of the following PHP code?

```

1. <?php
2. for ($i = 0; 0; $i++)
3. {
4.     print "i";
5. }
6. ?>

```

- a) infinite loop
- b) 0
- c) no output
- d) error

Answer: c

Explanation: The condition of the loop is always false 0.

2. What will be the output of the following PHP code?

```

1. <?php
2. for ($i = 0; $i < 5; $i++)
3. {
4.     for ($j = $i; $j > 0; $i--)
5.         print $i;
6. }
7. ?>

```

a) infinite loop

b) 0 1 2 3 4 5

c) 0 1 0 1 2 0 1 2 3 0 1 2 3 4 0 1 2 3 4 5 0 1 2 3 4 5

d) no output

Answer: a

Explanation: In the second loop j value is not being changed.

3. What will be the output of the following PHP code?

```

1. <?php
2. for ($i = 0; $i < 5; $i++)
3. {
4.     for ($j = $i; $j > $i; $i--)
5.         print $i;
6. }
7. ?>

```

a) infinite loop

b) 0 1 2 3 4 5

c) 0 1 0 1 2 0 1 2 3 0 1 2 3 4 0 1 2 3 4 5 0 1 2 3 4 5

d) no output

Answer: d

Explanation: The second loop does not execute as the check condition is always false.

4. What will be the output of the following PHP code?

```

1. <?php
2. $user = array("Ashley", "Bale", "Shrek", "Blank");
3. for ($x = 0; $x < count($user); $x++)
4. {
5.     if ($user[$x] == "Shrek")
6.         continue;
7.     printf ($user[$x]);
8. }

```

9. ?>

- a) AshleyBaleBlank
- b) AshleyBale
- c) AshleyBaleShrek
- d) No output

Answer: a

Explanation: Only the Shrek is skipped due to the continue statement.

5. What will be the output of the following PHP code?

```
1. <?php
2. $user = array("Ashley", "Bale", "Shrek", "Blank");
3.     for ($x=0; $x < count($user) - 1; $x++)
4.     {
5.         if ($user[$x++] == "Shrek")
6.             continue;
7.         printf ($user[$x]);
8.     }
9. ?>
```

- a) AshleyBaleBlank
- b) Bale
- c) AshleyShrek
- d) BaleBlank

Answer: a

Explanation: Only Bale is printed as \$x++ is done before printing and then checked.

6. What will be the output of the following PHP code?

```
1. <?php
2. $user = array("Ashley", "Bale", "Shrek", "Blank");
3. for ($x = 0; $x < count($user); $x)
4. {
5.     if ($user[$x++] == "Shrek")
6.         continue;
7.     printf ($user[$x]);
8. }
9. ?>
```

- a) AshleyBaleBlank
- b) BaleShrek
- c) AshleyBlank
- d) Bale

Answer: b

Explanation: x is incremented only inside loop i the if condition.

7. What will be the output of the following PHP code?

```
1. <?php
2. for ($i = 0; $i % ++$i; $i++)
3. {
4.     print "i";
5. }
6. ?>
```

- a) error
- b) infinite loop
- c) no output
- d) 0

Answer: b

Explanation: Loop condition is true as $i\%(i+1)$ is a float non zero value in php.

8. What will be the output of the following PHP code?

```
1. <?php
2. for ($i = 0; $i < 5; $i++)
3. {
4.     for (; $i < 5; $i++)
5.         print "i";
6. }
7. ?>
```

- a) iiii
- b) infinite loop
- c) iiii
- d) no output

Answer: a

Explanation: The i value is changed in the inner loop and reaches five, thus does not execute the second outer loop.

9. What will be the output of the following PHP code?

```
1. <?php
2. $a = array("hi", "hello", "bye");
3. foreach ($a as $value)
4. {
5.     if (count($a) == 2)
6.         print $value;
7. }
```


8. ?>

- a) hihellobye
- b) infinite loop
- c) hihello
- d) no output

Answer: d

Explanation: As `count($a)` returns 3 the condition is always false.

10. What will be the output of the following PHP code?

```
1. <?php
2. $a = array("hi", "hello", "bye");
3. for (;count($a) < 5;)
4. {
5.     if (count($a) == 3)
6.         print $a;
7. }
8. ?>
```

- a) ArrayArrayArrayArrayArrayArray....infinitely
- b) ("hi","hello","bye")("hi","hello","bye")("hi","hello","bye")("hi","hello","bye")...infinitely
- c) hihellobyehihellobyehihellobyehihellobyehihellobyehihellobye.....infinitely
- d) no output

Answer: a

Explanation: As `count($a)` returns 3 the condition is always true, thus it prints `$a`, which returns its data type.

1. What will be the output of the following PHP code?

```
1. <?php
2. function calc($price, $tax="")
3. {
4.     $total = $price + ($price * $tax);
5.     echo "$total";
6. }
7. calc(42);
8. ?>
```

- a) Error
- b) 0
- c) 42
- d) 84

Answer: c

Explanation: You can designate certain arguments as optional by placing them at the end of the list and assigning them a default value of nothing.

2. What will be the output of the following PHP code?

```
1.      <?php
2.      function a()
3.      {
4.          function b()
5.          {
6.              echo 'I am b';
7.          }
8.          echo 'I am a';
9.      }
10.     a();
11.     a();
12.     ?>
```

- a) I am b
- b) I am bI am a
- c) Error
- d) I am a Error

Answer: d

Explanation: This will be the output- I am a Fatal error: Cannot redeclare b()

3. What will be the output of the following PHP code?

```
1.      <?php
2.      function a()
3.      {
4.          function b()
5.          {
6.              echo 'I am b';
7.          }
8.          echo 'I am a';
9.      }
10.     b();
11.     a();
12.     ?>
```

- a) I am b
- b) I am bI am a
- c) Error
- d) I am a Error

Answer: c

Explanation: This will be the output- Fatal error: Call to undefined function b(). You cannot call a function which is inside a function without calling the outside function.

4. What will be the output of the following PHP code?

```
1.      <?php
2.      $op2 = "blabla";
3.      function foo($op1)
4.      {
5.          echo $op1;
6.          echo $op2;
7.      }
8.      foo("hello");
9.      ?>
```

- a) helloblabla**
- b) error**
- c) hello**
- d) helloblablablabla**

Answer: c

Explanation: If you want to put some variables in function that was not passed by it, you must use “global”. Inside the function type global \$op2.

5.What will be the output of the following PHP code?

```
1.      <?php
2.          function foo($msg)
3.          {
4.              echo "$msg";
5.          }
6.          $var1 = "foo";
7.          $var1("will this work");
8.      ?>
```

- a) error**
- b) \$msg**
- c) 0**
- d) will this work**

Answer: d

Explanation:It is possible to call a function using a variable which stores the function name.

6. What will be the output of the following PHP code?

```
1.      <?php
```

```
2.         echo "chr(52)";
```

```
3.         ?>
```

- a) 1
- b) 2
- c) 3
- d) 4

Answer: d

Explanation: The chr() function returns a character from the specified ASCII value. Since the ASCII value of 4 is 52, thus 4 was displayed.

7. What will be the output of the following PHP code?

```
1.         <?php
```

```
2.         echo ord ("hi");
```

```
3.         ?>
```

- a) 106
- b) 103
- c) 104
- d) 209

Answer: c

Explanation: The ord() function returns the ASCII value of the first character of a string. The ASCII value of h is 104, thus 104 was displayed.

8. What will be the output of the following PHP code?

```
1.         <?php
```

```
2.         echo (atan(0.50));
```

```
3.         ?>
```

- a) 0.11845976421345
- b) 0.23568451142521
- c) 0.46364760900081
- d) 1

Answer: c

Explanation: The atan() function returns the arc tangent of arg as a numeric value between -Pi/2 and Pi/2 radians.

9. What will be the output of the following PHP code?

```
1.         <?php
```

```
2.         define("GREETING","Hello you! How are you today?");
```

```
3.         echo constant("GREETING");
```

```
4.         ?>
```

- a) Hello you! How are you today?
- b) GREETING
- c) GREETING, Hello you! How are you today?
- d) "GREETING","Hello you! How are you today?"

Answer: a

Explanation: The define() function defines a constant.

10. What will be the output of the following PHP code?

```
1. <?php
2.     define("GREETING1","Hello you! How are you today?");
3.     define("GREETING2","Hello you! How are you today?");
4.     define("GREETING3","Hello you! How are you today?");
5.     echo defined("GREETING");
6. ?>
```

- a) 1**
- b) 0**
- c) 3**
- d) 4**

Answer: b

Explanation: The defined() function checks whether a constant exists.

1. What will be the output of the following PHP code?

```
1. <?php
2.     function sum($num1, $num2)
3.     {
4.         $total = $num1 + $num2;
5.         echo "chr($total)";
6.     }
7.     $var1 = "sum";
8.     $var1(5, 44);
9. ?>
```

- a) Error**
- b) 49**
- c) 1**
- d) Sum**

Answer: c

Explanation: It is possible to call a function using a variable which stores the function name also the chr() function returns a character from the specified ASCII value.

2. What will be the output of the following PHP code?

```
1. <?php
2.     function sum($num1, $num2)
3.     {
4.         $total = $num1 + $num2;
```

```
5.         echo "cos($total)";
6.     }
7.     sum(5,-5);
8. ?>
```

- a) 0
- b) 1
- c) 0.5
- d) -0.5

Answer: b

Explanation: cos() gives the cos value of the argument. Here the function returns 1.

3. What will be the output of the following PHP code?

```
1. <?php
2.     function b()
3.     {
4.         echo "b is executed";
5.     }
6.     function a()
7.     {
8.         b();
9.         echo "a is executed";
10.        b();
11.    }
12.    a();
13. ?>
```

- a) b is executedb is executedb is executed
- b) b is executeda is executed
- c) a is executed
- d) b is executeda is executedb is executed

Answer: d

Explanation: Simple order of execution.

4. What will be the output of the following PHP code?

```
1. <?php
2. function sum($x, $y)
3. {
4.     $z = $x + $y;
5.     return $z;
```

```

6. }

7. echo "5 + 10 = " . sum(7,13) . "<br>";

8. echo "7 + 13 = " . sum(2,4) . "<br>";

9. echo "2 + 4 = " . sum(5,10);

10. ?>

```

5. What will be the output of the following PHP code?

```

1. <?php
2. function addFive($num)
3. {
4.     $num += 5;
5. }
6. function addSix(&$num)
7. {
8.     $num += 6;
9. }
10. $orignum = 10;
11. addFive( &$orignum );
12. echo "Original Value is $orignum<br />";
13. addSix( $orignum );
14. echo "Original Value is $orignum<br />";
15. ?>

```

Answer: b

Explanation: addSix() passes value of the variable by reference.

6. What will be the output of the following PHP code?

```

1. <?php
2. function addFunction($num1, $num2)
3. {
4.     $sum = $num1 + $num2;
5.     return $sum;
6. }
7. $return_value = addFunction(10, 20);
8. echo "Returned value from the function : $return_value"
9. ?>

```

a) Returned value from the function : \$return_value

- b) Error
- c) Returned value from the function : 30
- d) Returned value from the function :

Answer: c

Explanation: Functions returns value 30.

7. What will be the output of the following PHP code?

```
1. <?php
2. function sayHello()
3. {
4.     echo "HelloWorld<br />";
5. }
6. $function_holder = "sayHello";
7. $function_holder();
8. ?>
```

- a) No Output
- b) Error
- c) sayHello
- d) HelloWorld

Answer: d

Explanation: It is possible to assign function names as strings to variables and then treat these variables exactly as you would the function name itself.

8. What will be the output of the following PHP code?

```
1. <?php
2. function one()
3. {
4.     echo " this works";
5.     function two()
6.     {
7.         echo "this too works";
8.     }
9. }
10. one();
11. two();
12. ?>
```

- a) error
- b) this works
- c) this worksthis too works
- d) this works this too works

Answer: c

Explanation: Two is declared in one and is called after one. Hence it works.

9. What will be the output of the following PHP code?

```
1. <?php
2. function do($myString)
3. {
4.     echo strpos($myString, "donkey",0);
5. }
6. do("The donkey looks like a horse.");
7. ?>
```

- a) 4
- b) 5
- c) 2
- d) None of the mentioned

Answer: a

Explanation: Donkey starts from position 4 in string.

10. What will be the output of the following PHP code?

```
1. <?php
2. function one()
3. {
4.     define("const","I am awesome!");
5.     echo constant("const");
6. }
7. one();
8. ?>
```

- a) I am awesome!!
- b) const
- c) const, I am awesome!!
- d) “const”,”I am awesome!”

Answer: a

Explanation: Using the define function to define the constant “const”.

1. What will be the output of the following PHP code?

```
1. <?php
2.     $title = "O'malley wins the heavyweight championship!";
3.     echo ucwords($title);
4. ?>
```

- a) O'Malley Wins The Heavyweight Championship!
- b) O'malley Wins The Heavyweight Championship!
- c) O'Malley wins the heavyweight championship!
- d) o'malley wins the heavyweight championship!

Answer: d

Explanation: The `ucwords()` function capitalizes the first letter of each word in a string. Its prototype follows: `string ucwords(string str)`.

2. What will be the output of the following PHP code?

```
1. <?php
2.     echo str_pad("Salad", 5)." is good.";
3. ?>
```

- a) SaladSaladSaladSaladSalad is good
- b) is good SaladSaladSaladSaladSalad
- c) is good Salad
- d) Salad is good

Answer: d

Explanation: The `str_pad()` function pads a string with a specified number of characters.

3. What will be the output of the following PHP code?

```
1. <?php
2.     $str = "Hello World"
3.     echo wordwrap($str,5,"<br>\n");
4. ?>
```

Answer: b

Explanation: The `wordwrap()` function wraps a string into new lines when it reaches a specific length.

4. What will be the output of the following PHP code?

```
1. <?php
2.     echo ucwords("i love my country");
3. ?>
```

- a) I love my country
- b) i love my Country
- c) I love my Country
- d) I Love My Country

Answer: d

Explanation: The `ucwords()` function converts the first character of each word in a string to uppercase.

5. What will be the output of the following PHP code?

```
1. <?php
2.     echo lcfirst("welcome to India");
3. ?>
```

- a) welcome to India
- b) welcome to india
- c) Welcome to India
- d) Welcome to india

Answer: a

Explanation: The lcfirst() function converts the first character of a string to lowercase.

6. What will be the output of the following PHP code?

```
1. <?php
2.     echo hex2bin("48656c6c6f20576f7226c6421");
3. ?>
```

- a) Hello World!
- b) welcome to india
- c) This is PHP!
- d) MCQ questons

Answer: a

Explanation: The hex2bin() function converts a string of hexadecimal values to ASCII characters.

7. What will be the output of the following PHP code?

```
1. <?php
2.     $str = addslashes('What does "yolo" mean?');
3.     echo($str);
4. ?>
```

- a) What does /"yolo/" mean?
- b) What does \\"yolo\\" mean?
- c) What does \"yolo\" mean?
- d) \\What does \"yolo\" mean?

Answer: c

Explanation: The addslashes() function returns a string with backslashes in front of predefined characters.

8. What will be the output of the following PHP code?

```
1. <?php
2.     $str = "Hello world. It's a beautiful day.";
3.     print_r (explode(" ", $str));
4. ?>
```

- a) Array ([0] => Hello [0] => world. [0] => It's [0] => a [0] => beautiful [0] => day.)
- b) Array ([0] => Hello [1] => world. [2] => It's [3] => a [4] => beautiful [5] => day.)
- c) Hello world. It's a beautiful day
- d) Array ([1] => Hello [2] => world. [3] => It's [4] => a [5] => beautiful [6] => day.)

Answer: b

Explanation: The explode() function breaks a string into an array.

9. What will be the output of the following PHP code?

```
1. <?php
2.     echo strstr("Hilla World","ia","eo");
3. ?>
```

- a) Hilla World
- b) Hello World
- c) ia
- d) eo

Answer: b

Explanation: The strstr() function translates certain characters in a string.

10. What will be the output of the following PHP code?

```
1. <?php
2.     echo strpos("I love php, I love php too!","PHP");
3. ?>
```

- a) 3
- b) 7
- c) 8
- d) 10

Answer: b

Explanation: The strpos() function finds the position of the first occurrence of a string inside another string.

1. What will be the output of the following PHP code?

```
1. <?php
2.     function A1($x)
3.     {
4.         switch($x)
5.         {
6.             case 1:
7.                 //this statement is the same as if($x == 1)
8.                 echo 'Case 1 was executed.';
9.                 break;
10.            case 2:
11.                //this statement is the same as if($x == 2)
12.                echo 'Case 2 was executed.';
13.                break;
14.            case 3:
15.                //this statement is the same as if($x == 3)
16.                echo 'Case 3 was executed.';
```

```

17.         break;
18.     case 4:
19.         //this statement is the same as if($x == 4)
20.         echo 'Case 4 was executed.';
21.         break;
22.     default:
23.         //this statement is the same as if $x does not equal the other conditions
24.         echo 'Default was executed.';
25.         break;
26.
27.     }
28. }
29. A1(9);
30. ?>

```

- a) Case 1 was executed
- b) Case 2 was executed
- c) Default was executed
- d) Case 4 was executed

Answer: d

Explanation: The switch statement is executed with \$x = 9.

2. What will be the output of the following PHP code?

```

1. <?php
2.     function uppercase($string)
3.     {
4.         echo ucwords($string);
5.     }
6.     $wow = "uppercase";
7.     $wow("Time to live king size");
8. ?>

```

- a) TIME TO LIVE KING SIZE
- b) Time to live king size
- c) Uppercase
- d) Time To Live King Size

Answer: d

Explanation: The ucwords() function converts the first character of each word in a string to uppercase.

3. What will be the output of the following PHP code?

```
1. <?php
2.     function TV($string)
3.     {
4.         echo "my favourite TV show is ".$string;
5.         function b()
6.         {
7.             echo " I am here to spoil this code";
8.         }
9.     }
10.    b();
11. ?>
```

- a) I am here to spoil this code
- b) Error
- c) My favourite TV show isI am here to spoil this code
- d) None of the mentioned

Answer: b

Explanation: b is undeclared if TV() is not called first.

4. What will be the output of the following PHP code?

```
1. <?php
2.     function TV($string)
3.     {
4.         echo "my favourite TV show is ".$string;
5.         function b()
6.         {
7.             echo " I am here to spoil this code";
8.         }
9.     }
10.    function b()
11.    {
12.        echo " I am here to spoil this code";
13.    }
14.    b();
15. ?>
```

- a) I am here to spoil this code
- b) Error
- c) my favourite TV show isI am here to spoil this code

d) None of the mentioned

Answer: a

Explanation: This one works because b is declared independent of TV() also.

5. What will be the output of the following PHP code?

```
1. <?php
2.     function TV($string)
3.     {
4.         echo "my favourite TV show is ".$string;
5.         function b()
6.         {
7.             echo " I am here to spoil this code";
8.         }
9.     }
10.    function b()
11.    {
12.        echo " I am here to spoil this code";
13.    }
14.    b();
15.    TV("Sherlock");
16. ?>
```

a) I am here to spoil this code

b) Error

c) My favourite TV show isI am here to spoil this code

d) None of the mentioned

Answer: b

Explanation: Function b is declared twice.

6. What will be the output of the following PHP code?

```
1. <?php
2.     function TV($string)
3.     {
4.         echo "my favourite TV show is ".$string;
5.         function b()
6.         {
7.             echo " I am here to spoil this code";
8.         }
```

```
9.      }  
  
10.     a("Sherlock");  
  
11.     b();  
  
12. ?>
```

- a) I am here to spoil this code
- b) Error
- c) my favourite TV show is SherlockI am here to spoil this code
- d) None of the mentioned

Answer: c

Explanation: b is declared as TV() is executed first.

7. What will be the output of the following PHP code?

```
1. <?php  
2.     function calc($num1, $num2)  
3.     {  
4.         $total = $num1 * $num2;  
5.     }  
6.     $result = calc(42, 0);  
7.     echo $result;  
8. ?>
```

- a) Error
- b) 0
- c) 42
- d) 84

Answer: a

Explanation: Function does not return anything.

8. What will be the output of the following PHP code?

```
1. <?php  
2.     function calc($num1, $num2)  
3.     {  
4.         $total = $num1 * $num2;  
5.         return $total;  
6.     }  
7.     $result = calc(42, 0);  
8.     echo $result;  
9. ?>
```

- a) Error
- b) 0

- c) 42
- d) 84

Answer: b

Explanation: Function returns \$total.

9. What will be the output of the following PHP code?

```
1. <?php
2.     $var = 10;
3.     function one()
4.     {
5.         echo $var;
6.     }
7.     one();
8. ?>
```

- a) Error
- b) 10
- c) No Output
- d) None of the Mentioned

Answer: c

Explanation: \$var is not global and hence is not available for one().

10. What will be the output of the following PHP code?

```
1. <?php
2.     function mine($m)
3.     {
4.         if ($m < 0)
5.             echo "less than 0";
6.         if ($ >= 0)
7.             echo "Not True";
8.     }
9.     mine(0);
10. ?>
```

- a) Less Than 0
- b) Not True
- c) No Output
- d) None of the Mentioned

Answer: b

Explanation: Argument is 0.

1. What will be the output of the following PHP code?

```
1. <?php
2. $x = 75;
3. $y = 25;
4. function addition()
5. {
6.     $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
7. }
8. addition();
9. echo $z;
10. ?>
```

- a) 100
- b) error
- c) 75
- d) 25

Answer: a

Explanation: z is a variable present within the \$GLOBALS array, it is also accessible from outside the function!

2. What will be the output of the following PHP code?

```
1. <?php
2. function 2myfunc()
3. {
4.     echo "Hello World";
5. }
6. 2myfunc();
7. ?>
```

- a) Hello World
- b) No Output
- c) ERROR
- d) None of the mentioned

Answer: c

Explanation: Function cannot begin with a number.

3. What will be the output of the following PHP code?

```
1. <?php
2. function _func()
3. {
4.     echo "Hello World";
5. }
```

```
6. _func();
```

```
7. ?>
```

- a) Hello World
- b) No Output
- c) ERROR
- d) None of the mentioned

Answer: a

Explanation: Function Beginning with “_” is valid.

4. What will be the output of the following PHP code?

```
1. <?php
2. function test($int)
3. {
4.     if ($int == 1)
5.         echo "This Works";
6.     if ($int == 2)
7.         echo "This Too Seems To Work";
8. }
9. test(1);
10. TEST(2);
11. ?>
```

- a) This Works
- b) This Too Seems To Work
- c) This WorksThis Too Seems To Work
- d) ERROR

Answer: c

Explanation: Function Is case Insensitive.

5. What will be the output of the following PHP code?

```
1. <?php
2. function mine($num)
3. {
4.     $num = 2 + $num;
5.     echo $num;
6. }
7. mine(3);
8. ?>
```

- a) 3
- b) \$num

- c) 5
d) None of the mentioned

Answer: c

Explanation: Simple arithmetic operation.

6. What will be the output of the following PHP code?

```
1. <?php
2. function mine($num)
3. {
4.     $num = 2 + $num;
5.     echo "$num";
6. }
7. mine(3);
8. ?>
```

- a) 3
b) \$num
c) 5
d) None of the mentioned

Answer: b

Explanation: The function is defined as echo "\$num". This means \$num is treated as a string and not as a variable.

7. What will be the output of the following PHP code?

```
1. <?php
2. function one($string)
3. {
4.     echo "I am ". $String;
5. }
6. one("Batman");
7. ?>
```

Answer: d

Explanation: Variable Undeclared) \$string is not the same as \$String.

8. What will be the output of the following PHP code?

```
1. <?php
2.     function string($title);
3.     {
4.         $title = ucwords($title);
5.         echo lcfirst($title);
6.     }
```

7. `string("you went full retard");`

8. `?>`

- a) You went full retard
- b) You Went Full Retard
- c) YOU WENT FULL RETARD
- d) you Went Full Retard

Answer: d

Explanation: ucwords() changes all the first letters to capitals. lcfirst() changes first letter of a string to small.

9. What will be the output of the following PHP code?

```
1. <?php
2.     function multi($num)
3.     {
4.         if ($num == 3)
5.             echo "I Wonder";
6.         if ($num == 7)
7.             echo "Which One";
8.         if ($num == 8)
9.             echo "Is The";
10.        if ($num == 19)
11.            echo "Correct Answer";
12.    }
13.    $scan = strpos("I love php, I love php too!", "PHP");
14.    multi($scan);
15. ?>
```

- a) I Wonder
- b) Which One
- c) Is The
- d) Correct Answer

Answer: b

Explanation: The strpos() function finds the position of the first occurrence of a string inside another string. In this case it returns 7.

10. What will be the output of the following PHP code?

```
1. <?php
2. function movie($int)
3. {
4.     $movies = array("Fight Club", "Kill Bill", "Pulp Fiction");
5.     echo "You Do Not Talk About ". $movie[$integer];
```

```
6. }  
7. movie(0);  
8. ?>
```

- a) You Do Not Talk About Fight Club
- b) You Do Not Talk About Kill Bill
- c) You Do Not Talk About Pulp Fiction
- d) None of the mentioned

Answer: a

Explanation: Simple use of arrays.

1. What will be the output of the following PHP code?

```
1. <?php  
2. function colour()  
3. {  
4.     $colors = array("red", "green", "blue", "yellow");  
5.     foreach ($colors as $value)  
6.     {  
7.         echo "$value <br>";  
8.     }  
9. }  
10. colour();  
11. ?>
```

2. What will be the output of the following PHP code?

```
1. <?php  
2. function addFunction($num1, $num2)  
3. {  
4.     $sum = $num1 + $num2;  
5.     return $sum;  
6. }  
7. $return_value = addFunction(10, 20);  
8. echo "Returned value from the function : " . $return_value  
9. ?>
```

- a) Returned value from the function : \$return_value
- b) Error
- c) Returned value from the function : 30
- d) Returned value from the function :

Answer: c

Explanation: Function returns 30.

3. What will be the output of the following PHP code?

```
1. <?php
2. function time($string)
3. {
4.     echo strstr("Towe Pa55", "ow5", $string);
5. }
6. time("ims");
7. ?>
```

- a) Time Pa55
- b) Towe Pa55
- c) Towe Pass
- d) Time Pass

Answer: d

Explanation: The strstr() function translates certain characters in a string.

4. What will be the output of the following PHP code?

```
1. <?php
2. function constant()
3. {
4.     define("GREETING", "Welcome to Narnia");
5.     echo greeting;
6. }
7. ?>
```

- a) Welcome to Narnia
- b) greeting
- c) GREETING
- d) ERROR

Answer: d

Explanation: By default constants are case sensitive. Hence an error will arise.

5. What will be the output of the following PHP code?

```
1. <?php
2. function constant()
3. {
4.     define("GREETING", "Welcome to Narnia",true);
5.     echo greeting;
6. }
```

7. ?>

- a) Welcome to Narnia
- b) greeting
- c) GREETING
- d) ERROR

Answer: a

Explanation: By default constants are case sensitive. But the third parameter in `define()`, if set to true, makes constants case insensitive.

6. What will be the output of the following PHP code?

```
1. <?php
2. function start($string)
3. {
4.     if ($string < 45)
5.         return 20;
6.     else
7.         return 40;
8. }
9. $t = start(90);
10. if ($t < 20)
11. {
12.     echo "Have a good day!";
13. }
14. else
15. {
16.     echo "Have a good night!";
17. }
18. ?>
```

- a) Have a good day!
- b) Have a good night!
- c) ERROR
- d) None of the mentioned

Answer: b

Explanation: Function returns 40. This is greater than 20, hence the output.

7. What will be the output of the following PHP code?

```
1. <?php
2. function case()
3. {
```



```
4.      ECHO "Hello World!<br>";
5.      echo "Hello World!<br>";
6.      EcHo "Hello World!<br>";
7.  }
8.  case();
9.  ?>
```

Answer: c

Explanation: Functions, keywords etc in php are case insensitive.

8. What will be the output of the following PHP code?

```
1. <?php
2. function email()
3. {
4.     $email = 'user@yahoo.com';
5.     $new = strstr($email, '@');
6.     echo $new;
7. }
8. email();
9. ?>
```

- a) user
- b) [\[email protected\]](#)
- c) @yahoo.com
- d) yahoo.com

Answer: c

Explanation: The strstr() function searches for the first occurrence of a string inside another string.

9. What will be the output of the following PHP code?

```
1. <?php
2. function string()
3. {
4.     echo strstr("Hello world!", 111);
5. }
6. string();
7. ?>
```

- a) o world!
- b) Hello world!
- c) 111
- d) No Output

Answer: a

Explanation: 111 is the ASCII value of o.

10. What will be the output of the following PHP code?

```
1. <?php
2. function CalAll($x,$y)
3. {
4.     echo ($x + $y);
5.     echo "<br>";
6.     echo ($x - $y);
7.     echo "<br>";
8.     echo ($x * $y);
9.     echo "<br>";
10.    echo ($x / $y);
11.    echo "<br>";
12.    echo ($x % $y);
13. }
14. $x = 10;
15. $y = 6;
16. CalcAll();
17. ?>
```