**1).What is the output of the following:**

**<?php  
$m = 3;  
$n = 0;  
function l() {  
   $m = 0;  
   $m++;  
   global $n  
   return array($n,$m);  
}  
echo implode((L(l())),',');  
?>**

0,1

Look out for PHP function names; they are not case sensitive, so both l() and L() will call the same method. The function doesn't have any declared parameters so the inside version of the function gets ignored.

#### 2). What would be output when the following code is run?

**<?php  
class M {  
   function m() {  
      echo "M";  
   }  
   function construct() {  
      echo "mmm";  
   }  
}  
$m = new M();  
$m->m();  
?>**

MM

#### 3). Which of the following could be used to add a book to an existing SimpleXMLElement object stored in $library representing a collection of books?



$library->addChild(new SimpleXMLElement("title", "Best book ever"))



$library->children()->add("title", "Best book ever")



$library->xpath(new SimpleXMLElement("title", "Best book ever"))



**$library->addChild("title", "Best book ever")**



$library->addBook("title", "Best book ever")

None of the other options are valid or would work

#### 4). What is the output of the following line of code:

**<?php  
$a = 4 << 2 + 1;  
echo $a;  
?>**

32

Look out for operator precedence here: http://www.php.net/manual/en/language.operators.precedence.php – the addition happens first, and then you do 4 << 3 = 32

#### 5). Which of the following is a valid way to pass the $callback parameter expected by array\_walk()? (choose three)



**An anonymous function**



An array containing the name of the function as the first element, and an array of arguments as the second element



An instantiated object followed by the object operator and the method to call



**An array containing an instantiated object as the first element, and the method name as the second element**



**A string containing the function name**



An array containing the function name as the first element

#### 6). How would you parse a web page at http://example.com/page.php as XML?

Top of Form



You can't; it must be well-formed XML.



DOMDocument::loadFile('http://example.com/page.php');



simple\_xml\_load\_html('http://example.com/page.php');



new SimpleXMLElement(file\_get\_contents('http://example.com/page.php'));



**DOMDocument::loadHTML(file\_get\_contents('http://example.com/page.php'));**

You can parse well-formed HTML using this method: http://php.net/manual/en/domdocument.loadhtml.php

#### 7). How many parameters does array\_merge() accept?

**Top of Form**

**As many as you want**

From php.net, the function prototype is: array array\_merge ( array $array1 [, array $... ] )

#### 8). What would you expect to get from PDOStatement::fetch() in its default mode?

Top of Form



Both an object and an array



**An array containing both associative and enumerated keys**



An associative array



An enumerated array



An error, you must pass in the fetch mode

The other answers are all things you could obtain using different fetch modes in PHP, but the default is PDO\_FETCH\_BOTH (http://php.net/manual/en/pdostatement.fetch.php). You can pass the fetch mode, but it isn't required.

#### 9). What is the output of the following code?

**<?php  
$pattern = '/[a-z]{4} /';  
$string = 'Mary had a little lamb';  
$matches = preg\_match($pattern, $string);  
print\_r($matches);  
?>**

1

Tricky question, because preg\_match() returns an integer – 1 if there were matches and 0 otherwise. To capture the matches, you pass a third parameter for them to be placed into (http://uk1.php.net/preg\_match). The pattern requires four lower case letters followed by a space, but doesn't anchor on to anthing so "ttle " does match.

#### 10). What is PDO::query() equivalent to?

Top of Form



execute(), fetch()



prepare(), fetch(), execute()



prepare(), fetch()



exec(), fetch()



**prepare(), execute()**

We usually use this when we don't have any parameters to pass in, for example fetching the contents of a lookup table.

#### 11). Which HTTP status code asks a user to provide credentials?

Top of Form



404



204



**401**



200



302

The 401 status code means "Not Authorised", so the user will be asked to identify themselves.

#### 12). What would be the output of this script?

**<?php  
ob\_start();  
echo "dreaming";  
$ob = ob\_get\_contents();  
echo strlen($ob);  
ob\_flush();  
?>**

dreaming8

We turn on output buffering, put "dreaming" into it, then grab the contents of the output buffer – but we haven't destroyed it. So when we echo strlen("dreaming"), an 8 goes into the output buffer as well, and then we flush it.

#### 13). What does status code 403 indicate?

Top of Form



**Forbidden**



Resource not found



Not modified



403 is not a valid status code



Moved permanently

Status code 403 means "Forbidden" – the user has provided credentials but still is not allowed to have acecss to this resource.

#### 14). What is the output of the following code?

**<?php  
$a = "a, b,c, d, e f, g";  
$b = array\_merge(explode(', ', $a), array("a", "b"));  
echo count($b);  
?>**

7

The explode function splits a string into an array, but the split is ", " (note the space after the comma). $a therefore has ", " in it four times, making five pieces ("b,c" and "e f" are two of the pieces. Then we add two more elements on the end (array\_merge doesn't mind duplicate entries) , making 7 in total.

#### 15). What is the output of this line of code?

**<?php  
echo 8 + 0x8 + 80 + 0x80;  
?>**

224

Hopefully you spotted that 0x8 is hexadecimal 8 ... which is the same as decimal 8. 0x80 is 8x16 (we use base 16 rather than base 10) which is 128. So 8+8+80+128 = 224

#### 16). Which of the following functions would be a valid way to create an array containing items from three existing arrays?



**array\_merge()**



array\_combine()



array\_splice()



array\_keys()



array\_intersect()

The others are all valid array functions but don't do what we need

#### 17). What is the output of the following line of code?

**<?php  
echo "4" + 05 + 011 + ord('a');  
?>**



An error



18



117



**115**



14

There are octal numbers in here, look for the leading zero on 05 and 011. octal 5 is the same as decimal 5, our string "4" will convert to a number when we try to add it to any other number, and 011 in octabl is (8 + 1) = 9. Hopefuly you knew that ord('a') was 97 and got 115 (ord('A') is 65, in case you're interested)

#### 18). What is the output of the following?

**<?php  
$a = 7;  
$b = 4;  
function b($a, $b) {  
   global $a, $b;  
   $a += 7;  
   $a++;  
   $b += $a;  
   return true;  
}  
echo $b, $a;  
?>**



1419



74



1519



1915



**47**

The function is never called.

#### 19). Which of the following are configuration settings for PHP? (choose two)



url\_fopen\_allow



url\_include\_allow



allow\_open\_url



**allow\_url\_include**



include\_fopen\_dir



open\_url\_dir



**open\_basedir**

None of the others are valid configuration directives for PHP <http://www.php.net/manual/en/ini.list.php>

#### 20). Is the following valid PHP code?

**<php>  
echo 'There's a worm in my apple';  
</php>**



Yes



**No**

Those style tags aren't valid for PHP.

#### 21). What is the output of:

**<?php  
$a = "0";  
echo strlen($a);  
echo empty($a) ? $a : 5;  
echo $a ?: 5;  
?>**



**105**



100



050



005



150

Strlen($a) is 1, then empty($a) is true, so we hit the first leg of the ternary and echo $a, which is zero. The third digit comes from the final line, $a is evaluated as false so we echo 5.

#### 22). What would happen when the following code was run?

**<?php  
define('Tree', 'oak');  
echo 'This tree is: ' . tree;  
?>**



A notice



This tree is: oak (and a notice)



An error



**This tree is: tree (and a notice)**



This tree is: (and a notice)

We define a constant but they are case-sensitive in PHP, so tree is an undefined constant. This is assumed to be a string in PHP (for historical reasons) but causes a notice to be emitted.

#### 23). Which of the following are valid constant names? (Choose three)



**R7yu**



9tur8UH4



8rutil



**AEN**



111



**ablwesc\_99**



YU-8

Valid constant names in PHP can be uppercase or lowercase, and contain numbers or underscores, but can't start with a number.

#### 24). What is the output of the following code?

**<?php  
$a = 42 & 05 + 17;  
echo $a;  
?>**



17



0



**2**



64



10

Did you spot the octal 5? It's irrelevant, because five is the same in base eight or base ten. The operator precedence means we do 17 + 5 = 22 first, then we do 42 & 22 ... & is a binary operation, so we need those numbers in binary first. 22 in binary: 10110 42 in binary: 101010 The only column where both numbers have a 1, is in the 2 place.

#### 25). What will the output of the following code be?

**<?php  
$a = range(3,9);  
foreach ($a as $b) {  
   switch($b) {  
      case 3:  
      $b = 7;  
      case 7:  
      $b = 3;  
      default:  
      // do nothing  
   }  
}  
echo implode('-',$a);  
?>**



7-4-5-6-7-8-9



3-4-5-6-7-8-9-10-11



**3-4-5-6-7-8-9**



7-4-5-6-3-8-9



7-4-5-6-3-8-9-10-11

The range(3,9) gives us an array containing all integers from 3 to 9. When we foreach over them, we can't change the values in the array, so the contents of $a remain unchanged (you need to use the array and the key to update a value during foreach)

#### 26). What is the output of:

**<?php  
$a = 10;  
echo strlen($a) . count($a);  
do {  
   echo $a . "elephpant ";  
   $a++;  
} while($a <= 1);  
?>**



**2110elephpant**



12



1210elephpant



21



10elephpan

The other answers are only partial. We start with 2 and 1 from the strlen and count of $a. Then we hit the do/while loop. This executes, but $a is NOT <=1 so the loop ends immediately.

#### 27). What is the output of this code:

**<?php  
function c($a, $b = 1, $c) {  
   return array($c, $a, $b);  
}  
list($a, $b, $c) = c(0,0,0);  
echo $b;  
?>**

0

First we call c() with three zero arguments, then we return those in a strange order (but they are all still zero). These get assigned into variables $a, $b and $c by the list() function and $b is zero.

#### 28). What is the output of the following code:

**<?php  
function a($a) {  
   echo $a . "&";  
}  
function b($a) {  
   echo "-" . $a;  
}  
$a = "!";  
$b = &$a;  
echo a(b($b));  
?>**



-!!&



!&-&



-&!



!&-!



**-!&**

This is a sharp question! We have $a as a string (containing symbols to make things more fun) and $b as a pointer to $a. We call b($b) which outputs "-!" and then we pass the result of that into a() ... but b() doesn't return anything so null gets passed into a() and only the & is echoed.

#### 29). Which of the following are superglobals in PHP? (choose three)



$\_FORMS



**$\_COOKIE**



$\_COOKIES



**$\_REQUEST**



$\_DATA



$\_ENVIRONMENT



**$\_SERVER**



$\_GLOBAL

Familiarity with PHP will help you on this one. More info on superglobals here: <http://www.php.net/manual/en/language.variables.predefined.php>

#### 30). Which of the following statements are true when applied to a Registry pattern? (choose two)



**Only one instance of the class can exist**



It implements ArrayAccess



**It is designed to store values of various types**



It uses static methods



It takes responsibility for instantiating objects

Applications will use a registry (or one of each of a number of related registry classes) to store values to be used multiple times in an application

#### 31). Is this statement true or false? "Methods declared as static must be called statically"



True



**False**

It's safe to call static methods dynamically, and you can do so – but the opposite is not true so you can only call non-static methods non-statically.

#### 32). What is the output of this code?

**<?php  
$wish\_list = array(1 => "Romeo and Juliet",  
   4 => "Bad Science",  
   2 => "To Kill A Mockingbird");  
print\_r(sort($wish\_list));  
?>**



Array ( [1] => Romeo and Juliet [4] => Bad Science [2] => To Kill A Mockingbird )



Array ( [1] => Romeo and Juliet [2] => To Kill A Mockingbird [4] => Bad Science )



**1**



Array ( [0] => Bad Science [1] => Romeo and Juliet [2] => To Kill A Mockingbird )



3

Look out for sort(), it (and its sister functions) return true! <http://www.php.net/sort>

#### 33). ArrayAccess is an example of a:



design pattern



**interface**



object



class

The ArrayAccess interface is part of the SPL <http://www.php.net/arrayaccess>

#### 34). Using the notation self::$property refers to:



A property of the current object



The $property of the current object



The class constant $property in this class



**The class property $property in this class**



A variable called $property in this class or any parent clas

The scope resolution operator (the two colons ::) indicates this is a class property, and self means it's the property of the current class.

#### 35). Which of the following is a valid namespace operator in PHP?



**Ans \**



::



@



/



->

We use the backslash as the namespace operator in PHP <http://www.php.net/namespaces>

#### 36). What is the output of the following code?

**<?php**

**$x = 1;**

**function print\_conditional() {**

**static $x;**

**if($x++ == 1)**

**echo "many";**

**echo "good";**

**echo "things";**

**return $x;**

**}**

**print\_conditional();**

**$x++;**

**print\_conditional();**

**?>**



manygoodthingsmanygoodthings



manygoodthings



goodthingsgoodthings



**goodthingsmanygoodthings**



there is no output

The $x outside the function has no link with $x inside the function. $x inside the function is a static variable so it retains its value between function calls. The first time we call print\_conditional(), $x is static but undefined. The $x++ post increment operator returns the old value, then increments the variable, so on the first pass, the if() returns false and then $x is 1. Look very closely at that if! There are no { } brackets so only the "many" line is conditional, the "good" and "things" lines will run regardless (this is why indentation should follow program flow). On the second call, $x is 1 so the if() evaluates to true and we get the "many" line as well.

#### 37). What does the html\_errors configuration directive do?



Enable reporting of invalid HTML output



Show PHP errors on webpages



html\_errors is not a valid configuration directive



**Apply HTML formatting to PHP errors**



Suppress browser errors

The html\_errors setting enables hyperlinks and formatting in any error output; other libraries (for example Xdebug) may also observe this setting.

#### 38). What is the output of the following code?

**<?php**

**$a = 1;**

**function calculate() {**

**global $a;**

**$a += 7;**

**$a = $a \* 043;**

**return --$a;**

**}**

**echo $a;**

**?>**



342



279



0



343



**1**

The function is never called ...

#### 39). What is the output of the following code?

**<?php**

**function format(&$item) {**

**$item = strtoupper($item) . '.';**

**return $item;**

**}**

**$shopping = array("fish", "bread", "eggs", "jelly", "apples");**

**array\_walk($shopping, "format");**

**$shopping = sort($shopping);**

**echo $shopping[1];**

**?>**



"APPLES."



"apples"



"BREAD."



**nothing (no output)**



"fish"

This is the sort() function returning true, which makes no sense with array notation so PHP doesn't return us anything at all.

#### 40). What is the output of the following code?

**<?php**

**class Content**

**{**

**public function publish() {**

**$this->published = true;**

**$this->article();**

**return true;**

**}**

**protected function article() {**

**echo "<i>Article:</i>";**

**}**

**}**

**class Article extends Content**

**{**

**public function article() {**

**echo "<i>Post:</i>";**

**}**

**}**

**$post = new Article();**

**echo $post->publish();**

**?>**



<i>Post:</i>



an Error



**<i>Post:</i><i>Post:</i>1**



<i>Article:</i>



<i>Post:</i><i>Article:</i>1

We have a class Content and another class Article which extends Content. When we instantiate a new Article() the function article() becomes our constructor because the method name meets the class name (this is from PHP 4 days but is still true) so we echo "*Post:*". Then we call publish() on our object, which calls Article::article() again (NOT Content::article()), and returns true. We echoed the output of our call and the boolean becomes a 1 when we echo it.

#### 41). Given the following code:

**<?php**

**Interface Verifiable**

**{**

**public function verify();**

**}**

**Class Cheque**

**{**

**public function verify() {**

**// interesting stuff happens**

**return true;**

**}**

**}**

**Class CurrencyCheque extends Cheque implements Verifiable {**

**}**

**?>**

#### What happens when we instantiate a CurrencyCheque object?



An warning because the interface isn't implemented



**A new CurrencyCheque object is created**



The method must be redefined in CurrencyCheque



An error that the interface must be defined in the parent object



A fatal error

Look closely at that parent object – it already defines the method we need to satisfy the interface. Therefore we can implement it in the child class without needing to redeclare the method – everything will work fine.

 Solve it

#### 42). What is the output of the following code?

**<?php**

**if(strcmp("hi", "HI")) echo "hello";**

**elseif(strcasecmp("hi","HI")) echo "world";**

**else throw new Exception("HI");**

**?>**



"world"



an Exception



nothing (no output)



**"hello"**



an error

The key here is knowing that strcmp is case-sensitive but that it returns zero if things match http://www.php.net/strcmp. For anything that isn't zero, the "if" will evaluate to true.

#### 43). How can you recover the original information from this string?

**a:4:{i:2;s:3:"foo";i:3;s:4:"spot";i:4;s:6:"stripe";s:3:"bar";i:64;}**



using json\_decode()



using the mcrypt extension



**using unserialize()**



using a database library



you can't – this is a custom format

This data is PHP's serialized format so you wamt to unserialize it – it does look a bit like JSON, but the clue is that the example here includes information about the data type, which JSON does not.

#### 44). Which of the following are true (choose three)?



**Anonymous functions can accept variables passed into them at call time**



Anonymous functions cannot be called recursively



**Closures are a kind of anonymous function**



Anonymous functions can have function names



It is possible to return multiple values from an anonymous function



**Closures can have variable values "baked" in at declare time**



Anonymous functions are always created in the global scope

Anonymous functions were new in PHP 5.3, however hopefully either experience or study means you were able to pick your way through this question. More information on anonymous functions: <http://www.php.net/manual/en/functions.anonymous.php>

#### 45). What is the output of the following?

**<?php**

**echo chr((ord('a') + ord('A'))/2);**

**?>**

Q

Take the time to work this one out – it looks tiny but it still needs some attention. First we take ord('a') = 97 and ord('A') = 65, add them together (162) and then divide by 2 to make 81. Now count forwards from 65 (B is 66, C is 67, and so on) until you reach 81. The chr() function is the opposite of ord() and makes numbers back into letters.

#### 46). How would you efficiently extract data from a csv file which is several gigabytes in size?



Use a custom stream wrapper



file\_get\_contents() and explode()



**fopen() and fgetcsv()**



file() and explode()



**fopen(), fgets() and explode(**

Either of these solutions would work pretty well – for a very large file, loading it into memory with file\_get\_contents or something similar would make PHP run out of memory (or at least use up a lot of it!) so it's better to use a file-pointer-based approach and do it a line at a time.

#### 47). Which of the following would you use to validate incoming data from a web form? (choose three)?



**filter\_\* functions**



stripslashes()



htmlspecialchars()



database escape functions



**ctype\_\* functions**



strip\_tags()



**preg\_\* functions**



url\_decode()

The key word in the question is "validate" – some of the options to choose from are great ways to sanitize data, but not to validate it. The three answers are all great ways to work with incoming data which you should be familiar with: http://www.php.net/manual/en/book.filter.php http://www.php.net/manual/en/book.ctype.php http://www.php.net/manual/en/ref.pcre.php

#### 48). Which of the following php configuration directives were deprecated in PHP 5.3 (tick as many as apply)?



y2k\_compliance



**safe\_mode**



disable\_functions



**register\_globals**



gpc\_order



max\_file\_uploads



register\_syslog



**register\_long\_arrays**

#### 49). What is the output of the following?

**<?php**

**$a = 0xf2 + 0x09;**

**$b = $a >> 3;**

**echo $b;**

**?>**



**31**



0



27



7



16

There are three steps here. First, take the hexadecimal numbers and convert them to decimal, which gives you (242 + 9) = 251. Next, write 251 in binary (it's 255 less 4 if you like shortcuts) which is 11111011. Now shift that number to the right three steps, ignoring the digits which get shifted away to the right of the 1 column. This should give you 11111 – which is 31 when you turn it back into decimal.

#### 50). Which of the following would allow you to send a POST request to a remote resource via file\_get\_contents()?



You can't, use the curl extension or an equivalent



stream\_context(array("method" => "POST"));



array("method"=>"post");



stream\_context\_create("http\_method" => "POST"));



**stream\_context\_create(array("http" => array("method" => "POST")));**

It is possible to make all kinds of requests and set any headers you like by setting the context on a stream. Here, we're setting the method option inside the http element of the context.

#### 51). Which function would you use to re-order an array by its keys?

Ksort

There are lots of ways of sorting arrays, by key or by value, keeping keys intact or not – this time you need ksort. http://www.php.net/ksort

#### 52). What does the chr() function do?



**Returns the ascii code of the given character**



Returns the ascii codes for all characters in the string as an array



Returns the character at a given offset



Returns the character for a given ascii code



Returns the number of characters in the string

The chr() is the opposite of the ord() function. <http://www.php.net/chr>

#### 53). Given this code sample:

**<?php**

**interface A {}**

**class C {}**

**class B extends C {}**

**class E extends C implements A {}**

**class D extends E{}**

**$b = new B();**

**$e = new E();**

**$c = new C();**

**$a = new B();**

**$d = new D();**

**?>**

#### Which of the following statements are true? (choose 4)



$c instanceof B



**$d instanceof A**



**$d instanceof C**



$e instanceof B



**$e instanceof A**



**$c instanceof C**



$a instanceof E

This question is really about polymorphism. Objects will claim to be instances of the classes they are actually instances of, but will also identify themselves as any of their ancestors or as any interface that their class, or any ancestor class, implements.

#### 54). Which of the following session save handlers are available by default in PHP? (choose 3)



redis



postgresql



**files**



**memcache**



AWS



foxpro



**sqlite**

PHP defaults to files, and many other options are available but providing you have memcache or sqlite, either of these can be use for session storage

#### 55). The Active Record design pattern is used for which of the following?



Creating audit trails of all file changes



**Easily storing and retrieving objects in the database**



Integrating PHP and Ruby on Rails projects



Separating business and presentation logic within a system



Managing output handlers

Made famous by ruby on rails, the active record pattern is common in model layers to handle the translation between the objects we use in applications and the way we store them.

#### 56). With a single existing cookie set for this domain with the key "theme" and the value "green", what does the following code output?

**<?php**

**print\_r($\_COOKIE);**

**setcookie('theme', NULL, time() - 3600);**

**print\_r($\_COOKIE);**

**unset($\_COOKIE);**

**print\_r($\_COOKIE);**

**?>**



an error



Array ( [theme] => green )



**Array ( [theme] => green ) Array ( [theme] => green )**



Array ( [theme] => green ) Array ( [theme] => green ) Array ( [theme] => green )



nothing

Here we see the contents of the $\_COOKIE array, but setting another cookie will not make any difference until the next request. The contents of $\_COOKIE are output again, then we unset the array. We haven't destroyed any of the cookies being exchanged, but we have removed them from that variable in our script. So we don't see the output (but there would be a notice if they were enabled).

#### 57). What does the following code output?

**<?php**

**$i = function ($j) {**

**$i = $j + 4;**

**$i++;**

**return $i;**

**};**

**$j = 6;**

**echo $i($j);**

**?>**



10



syntax error



nothing



**11**



6

In this script, we have an anonymous function. We declare it and assign it to $i. Then we call it, and pass in $j which is set to 6. The $i inside the scope of the function has no link with anything outside, so it's fine (although confusing!) to use the variable name there. 6 +4 is 10, then we increment $i and make 11, then return it.

#### 58). What is the output of the following code?

**<?php**

**function swings(&$park) {**

**$park++;**

**$park = roundabout($park);**

**}**

**function roundabout($park) {**

**$park \*= 2;**

**}**

**$park = 17;**

**echo swings($park);**

**?>**



19



37



36



74



**nothing**

Nothing is returned by swings() so there's no output echoed.

#### 59). What is the output of the following code?

**<?php**

**$x = 1;**

**function print\_conditional($x) {**

**if($x++ == 1)**

**echo "none";**

**echo "one";**

**echo "none";**

**return $x;**

**}**

**print\_conditional($x);**

**$x++;**

**print\_conditional($x);**

**?>**



onenone



**noneonenoneonenone**



nonenoneonenone



onenoneonenoneonenone



noneonenone

This is really similar to an earlier question but don't be fooled; there are some differences. $x is 1 and we call print\_conditional, passing in $x. The post increment means the if is evaluated to true and all three echo lines will be executed. The function returns $x but it isn't assigned so $x is still 1. In the last two lines we increment $x and pass it into the function again, but this time the if does not evaluate to true. The absence of curly braces however means that we do still get the "one" and second "none" line.

#### 60). Which object method is automatically called when an object is cloned?

Top of Form



\_\_copy()



\_\_wakeup()



\_\_drone()



**\_\_clone()**



\_\_call()

This question is about magic methods; these are a key element of OOP theory in PHP. <http://www.php.net/manual/en/language.oop5.magic.php>

#### 61). What is the output of the following code?

**<?php**

**$g = range(5,8);**

**$h = array("a", "b", "c", "e");**

**for($i = 0; $i < count($g); $i++) {**

**foreach ($h as $j) {**

**echo $i.$j;**

**break;**

**}**

**}**

**?>**



**0a1a2a3a**



5a6a7a8a



0a0b0c0e



0a0b0c0e1a1b1c1e2a2b2c2e3a3b3c3e



5a5b5c5e6a6b6c6e7a7b7c7e8a8b8c8e

We start off with the range statement which produces an array containing the values: 5,6,7,8, used only for count() purposes, and another array with letters in. The for loop has a foreach inside it, but with a break statement, so the for loop operates as normal but the foreach loop only gets run once before we break out of it, on each turn around the for loop.

#### 62). What is the output of the following code?

**<?php**

**$s = "This sentence contains many words";**

**$r = explode(' ', ucfirst($s));**

**sort($r);**

**echo implode(',', $r);**

**?>**



This Sentence Contains Many Words



**This,contains,many,sentence,words**



This,Sentence,Contains,Many,Words



This contains many sentence words



This Contains Many Sentence Words

Applying ucfirst to $s makes no difference, but exploding on a space splits the sentence into an array with one word in each element. We sort the words but look out because sort is case-sensitive and will sort the capital letter first, then the rest alphabetically.

#### 63). Given a class called SoapFunctions and a working WSDL for the methods in that class, what needs to be added to the code below to serve those methods over SOAP?

**<?php**

**require("SoapFunctions.php");**

**$s = new SoapServer($wsdl);**

**$s->handle();**

**?>**



$s->setHandler("SoapFunctions");



set\_soap\_class($s, "SoapFunctions");



set\_soap\_handler($s, "SoapFunctions");



**$s->setClass("SoapFunctions");**



nothing to add, the code above would work

None of the other options exist or are valid; the missing line needs the setClass() call. <http://www.php.net/manual/en/soapserver.setclass.php>

#### 64). Which of the following is a magic method in PHP 5.3? (choose three)

Top of Form



\_\_walk()



**\_\_sleep()**



\_\_return()



**\_\_call()**



\_\_function()



\_\_add()



**\_set()**

Check out the page on magic methods to see a full list: <http://www.php.net/manual/en/language.oop5.magic.php>

#### 65). Which function would transform the string "excellent PHP functions" into the string "Excellent PHP Functions"?

ucwords

This method will make the first letter of every word into an upper case letter, regardless of its previous case.

#### 66). Which of the following would offer protection against an SQL injection attack? (choose two)



**mysql\_real\_escape\_string()**



mysql\_true\_escape\_string()



addslashes()



**using PDO and prepared statements**



htmlspecialchars()



strip\_tags()

One of the answers here isn't a real PHP function (option B, mysql\_true\_escape\_string) and the others aren't effective against SQL attacks (although they're great in other security-related uses)

#### 67). What is the output of the following code?

**<?php**

**class M {**

**public function identify() {**

**echo self::myName();**

**}**

**public function myName() {**

**return "Mike";**

**}**

**}**

**class N extends M {**

**public function myName() {**

**return "November";**

**}**

**}**

**function m(N $n) {**

**$n->identify();**

**}**

**$m = new N();**

**m($m);**

**?>**



Fatal error



Catchable fatal error



**Mike**



November



Syntax error

Why is the answer C, not D (or any of the others). Put simply: late static binding http://php.net/manual/en/language.oop5.late-static-bindings.php What happens is that we instantiate an N confusingly stored in $m, and then call m() on it which, after passing the typehint successfully, calls the identify() method in the N class. This is inherited from M, but PHP will use M where we put "self" nstead of using N. Late static binding means we could use "static" instead of "self" and get "November" as the output.

#### 68). What is the output of the following code?

**<?php**

**$a = "Apple";**

**echo <<<'A'**

**pass me that $a**

**A;**

**?>**



syntax error



"pass me that Apple"



"pass me that apple"



fatal error



**pass me that $a**

This question shows a NOWDOC rather than a HEREDOC, so the variables won't be interpreted http://www.php.net/manual/en/language.types.string.php#language.types.string.synta x.nowdoc

#### 69). How would you change a SimpleXMLElement object into a DOMElement? (choose two)



convert SimpleXMLElement to XML, then create a DOMElement from the XML



using simplexml\_export\_dom()



**using dom\_import\_simplexml()**



SimpleXMLElement::asDom()



SimpleXMLElement::saveXML()

If you didn't remember the name of the function, perhaps you can eliminate some of the other options. Options B, D and E are nonsense and don't exist in PHP (although DOMDocument has a saveXML() method). Option A looks plausible but you don't create OMElement from XML – you create a DOMDocument. So go for option C which is exactly what is needed: <http://www.php.net/manual/en/function.dom-import-simplexml.php>