

# Zend Certification Preparation Pack

## ZCE 5.5

70 sample questions, with answers and explanations – plus information about what to expect on the day.

*by Lorna Mitchell*

# Zend Certification Preparation Pack

Lorna Mitchell

This book is for sale at <http://leanpub.com/zce>

This version was published on 2015-02-11



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2015 Lorna Mitchell

# **Tweet This Book!**

Please help Lorna Mitchell by spreading the word about this book on [Twitter!](#)

The suggested tweet for this book is:

Getting ready for my ZCE exam with the Zend Preparation Pack by @lornajane #zceprep

The suggested hashtag for this book is [#zceprep](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#zceprep>

## Also By **Lorna Mitchell**

[N Ways To Be A Better Developer](#)

[Git Workbook](#)

# Contents

About These Questions . . . . .	1
About The Author . . . . .	1
Preparing for ZCE . . . . .	1
What To Expect On Exam Day . . . . .	2
<b>Questions Set for ZCE 5.5 . . . . .</b>	<b>3</b>
Question 1 . . . . .	3
Question 2 . . . . .	3
Question 3 . . . . .	4
Question 4 . . . . .	4
Question 5 . . . . .	5
Question 6 . . . . .	5
Question 7 . . . . .	5
Question 8 . . . . .	6
Question 9 . . . . .	6
Question 10 . . . . .	6
Question 11 . . . . .	7
Question 12 . . . . .	7
Question 13 . . . . .	7
Question 14 . . . . .	8
Question 15 . . . . .	8
Question 16 . . . . .	8
Question 17 . . . . .	9
Question 18 . . . . .	9
Question 19 . . . . .	9
Question 20 . . . . .	10
Question 21 . . . . .	10
Question 22 . . . . .	11
Question 23 . . . . .	11
Question 24 . . . . .	12
Question 25 . . . . .	12
Question 26 . . . . .	12
Question 27 . . . . .	13
Question 28 . . . . .	13
Question 29 . . . . .	13
Question 30 . . . . .	14
Question 31 . . . . .	14
Question 32 . . . . .	15

## CONTENTS

Question 33 . . . . .	15
Question 34 . . . . .	16
Question 35 . . . . .	16
Question 36 . . . . .	16
Question 37 . . . . .	17
Question 38 . . . . .	17
Question 39 . . . . .	17
Question 40 . . . . .	18
Question 41 . . . . .	18
Question 42 . . . . .	19
Question 43 . . . . .	19
Question 44 . . . . .	20
Question 45 . . . . .	20
Question 46 . . . . .	21
Question 47 . . . . .	21
Question 48 . . . . .	21
Question 49 . . . . .	22
Question 50 . . . . .	22
Question 51 . . . . .	22
Question 52 . . . . .	23
Question 53 . . . . .	24
Question 54 . . . . .	24
Question 55 . . . . .	25
Question 56 . . . . .	25
Question 57 . . . . .	25
Question 58 . . . . .	26
Question 59 . . . . .	26
Question 60 . . . . .	27
Question 61 . . . . .	27
Question 62 . . . . .	28
Question 63 . . . . .	28
Question 64 . . . . .	29
Question 65 . . . . .	29
Question 66 . . . . .	29
Question 67 . . . . .	30
Question 68 . . . . .	30
Question 69 . . . . .	31
Question 70 . . . . .	31
<b>Sample Answers To Sample Questions . . . . .</b>	<b>32</b>
Question 1 . . . . .	32
Question 2 . . . . .	32
Question 3 . . . . .	33
Question 4 . . . . .	33
Question 5 . . . . .	34
Question 6 . . . . .	34

## CONTENTS

Question 7 . . . . .	35
Question 8 . . . . .	35
Question 9 . . . . .	35
Question 10 . . . . .	36
Question 11 . . . . .	36
Question 12 . . . . .	36
Question 13 . . . . .	37
Question 14 . . . . .	38
Question 15 . . . . .	38
Question 16 . . . . .	39
Question 17 . . . . .	39
Question 18 . . . . .	40
Question 19 . . . . .	40
Question 20 . . . . .	40
Question 21 . . . . .	41
Question 22 . . . . .	42
Question 23 . . . . .	42
Question 24 . . . . .	42
Question 25 . . . . .	43
Question 26 . . . . .	43
Question 27 . . . . .	44
Question 28 . . . . .	44
Question 29 . . . . .	45
Question 30 . . . . .	46
Question 31 . . . . .	46
Question 32 . . . . .	47
Question 33 . . . . .	47
Question 34 . . . . .	48
Question 35 . . . . .	48
Question 36 . . . . .	48
Question 37 . . . . .	49
Question 38 . . . . .	49
Question 39 . . . . .	49
Question 40 . . . . .	50
Question 41 . . . . .	51
Question 42 . . . . .	51
Question 43 . . . . .	52
Question 44 . . . . .	52
Question 45 . . . . .	53
Question 46 . . . . .	53
Question 47 . . . . .	54
Question 48 . . . . .	54
Question 49 . . . . .	54
Question 50 . . . . .	55
Question 51 . . . . .	55
Question 52 . . . . .	56

## CONTENTS

Question 53 . . . . .	57
Question 54 . . . . .	57
Question 55 . . . . .	58
Question 56 . . . . .	58
Question 57 . . . . .	59
Question 58 . . . . .	59
Question 59 . . . . .	60
Question 60 . . . . .	61
Question 61 . . . . .	61
Question 62 . . . . .	61
Question 63 . . . . .	62
Question 64 . . . . .	62
Question 65 . . . . .	63
Question 66 . . . . .	63
Question 67 . . . . .	63
Question 68 . . . . .	64
Question 69 . . . . .	65
Question 70 . . . . .	65



Being certified proves that you have a command of the topics included in the exam, and is becoming increasingly recognised around the world by employers. Even if you aren't looking to change jobs, studying for ZCE means covering a huge amount of PHP content in depth – and that is good for any developer.

## About These Questions

This pack of questions is intended as a mock ZCE exam, giving a similar style of question and answer that you might find on the day. They are NOT the real exam question, nor written by the authors of the other exam questions, however these practice questions are intended to help you get used to applying your knowledge to the style of question you will find in the ZCE itself.

## About The Author

Lorna Mitchell (@lornajane) is ZCE for PHP 5.0/5.3/5.5 and has taught hundreds of students ZCE over the last few years, on her public and private courses, and virtually to students around the world. She is the author of various PHP books and a regular conference speaker. Lorna writes prolifically for a number of outlets, mostly about PHP, and you can find out more about her and read many hundreds of her blog posts on her site <http://www.lornajane.net>.

## Preparing for ZCE

1. Investigate what is involved – there is a list of syllabus topics at <https://www.zend.com/services/certification/php-5-certification/> so take a look there and figure out if that sounds like something you're interested in. Usually I recommend 2 years experience in PHP programming before attempting the ZCE.
2. Research any areas where the questions seemed tricky for you. It's likely that there are some areas of PHP that you haven't been using regularly, so you'll need to study. Also remember that there aren't any reference materials or internet access allowed in the exam, so it might be helpful to make some flashcards or other reminders to help you retain all that information. I curate a bundle of links that I've recommended to ZCE students: <http://lrnja.net/LUzcMp> and this is actively maintained so let me know if any of the links are dead or there are resources you'd recommend to others
3. Prepare with a course. I've been teaching ZCE for a lot of years now and I rarely teach anyone who doesn't already know a lot of PHP. Running the ZCE preparation classes (I provide these, so does Zend, there may be others local to you) gives me a chance to really teach anything which someone hasn't used before, and recap on any tricky questions that come up. I love to teach, but the sheer size of the syllabus makes this easily one of the most challenging classes that comes up for me – and I love it.
4. Book the exam at a centre near you. If you booked a preparation class, then book the exam very shortly after that, because when else will you spend a number of days uninterrupted just thinking about ZCE topics? Don't let confidence put you off.
5. Practice with some questions and make sure you know what to expect in the exam itself (more tips on the next page and in Zend's study guide).
6. Take the exam – I wish you the very best of luck!

## What To Expect On Exam Day

The exam is taken on a computer, at a Pearson Vue test centre (these are everywhere, the organisation administers many different kinds of exam). You will not be allowed to take anything into the exam with you – no phone, no drinks, nothing – and there is no access to any reference material. I usually recommend people beg for a glass of water if they will need it!

You will be allowed 90 minutes maximum for the exam, and there will be time-keeping devices visible. The exam consists of approximately 70 questions. Why approximately? Because some are weightier than others, so you might have 69 questions because one was really tough, or 71 because there were a few too many little ones in there!

70 questions in 90 minutes is about one question per minute with a bit of time set aside for any panicking that might be needed towards the end :)

You will be given something to write on and something to write with. Sometimes that's pencil and paper, but I have also seen laminated sheets with wipe-clean pens given. Use these to your advantage and write down the steps as you work through a tricky question.

Don't be tempted to rush through the questions. The software that runs the exam will let you flag up questions that you want to come back to later. You begin at question one, and move on through the test. When you get to the end you will be presented with a screen displaying a grid of all the questions, with indications for which ones you flagged up, and also showing any that you didn't answer. This makes it very easy to go through and "mop up" anything that you didn't get the first time.

Do not rush through the questions. Seriously.

Many of the questions rely on some small detail of the code, and you can't run it to check what will happen. Rushing might mean you miss the detail, so take it at a steady pace. We all get a bit panicky in exams, so write down as many intermediate steps as you need to in order to keep track of the more complicated questions – and especially use a pen for the binary/octal/hexadecimal ones!

If you think you are finished, just press the final button – your result will show on screen immediately (I wish someone had told me that the first time around, it was a shock!) and the test centre will be able to give you a printout showing your result (you won't get a mark, it's just a pass or fail).

The rest is just common sense – try to get a good night's sleep the night before, arrive in plenty of time, have something to eat beforehand, and don't forget to keep breathing!

Good luck :)

# Questions Set for ZCE 5.5

70 questions in the style of the Zend Certified Engineer exam.

## Question 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())),'');
```

- A. 0, 5
- B. 0, 4
- C. An error
- D. 0,1
- E. 4,1

[Jump to answer](#)

## Question 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();
```

[textbox]

[Jump to answer](#)

## Question 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?

- A. \$library->addChild(new SimpleXMLElement("title", "Best book ever"))
- B. \$library->children()->add("title", "Best book ever")
- C. \$library->xpath(new SimpleXMLElement("title", "Best book ever"))
- D. \$library->addChild("title", "Best book ever")
- E. \$library->addBook("title", "Best book ever")

[Jump to answer](#)

## Question 4

What is the output of the following line of code:

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

- A. 9
- B. 16
- C. 17
- D. 1
- E. 32

[Jump to answer](#)

## Question 5

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

- A. An anonymous function
- B. An array containing the name of the function as the first element, and an array of arguments as the second element
- C. An instantiated object followed by the object operator and the method to call
- D. An array containing an instantiated object as the first element, and the method name as the second element
- E. A string containing the function name
- F. An array containing the function name as the first element

[Jump to answer](#)

## Question 6

How would you parse a web page at <http://example.com/page.php> as XML?

- A. You can't; it must be well-formed XML.
- B. `DOMDocument::loadFile('http://example.com/page.php');`
- C. `simple_xml_load_html('http://example.com/page.php');`
- D. `new SimpleXMLElement(file_get_contents('http://example.com/page.php'));`
- E. `DOMDocument::loadHTML(file_get_contents('http://example.com/page.php'));`

[Jump to answer](#)

## Question 7

How many parameters does array\_merge() accept?

- A. As many as you want
- B. 3
- C. 2
- D. 1
- E. 0

[Jump to answer](#)

## Question 8

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

- A. Both an object and an array
- B. An array containing both associative and enumerated keys
- C. An associative array
- D. An enumerated array
- E. An error, you must pass in the fetch mode

[Jump to answer](#)

## Question 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);
```

[textbox]

[Jump to answer](#)

## Question 10

What is `PDO::query()` equivalent to?

- A. `execute()`, `fetch()`
- B. `prepare()`, `fetch()`, `execute()`
- C. `fetch()`, `execute()`
- D. `exec()`, `fetch()`
- E. `prepare()`, `execute()`, `fetch()`

[Jump to answer](#)

## Question 11

Which HTTP status code asks a user to provide credentials?

- A. 404
- B. 204
- C. 401
- D. 200
- E. 302

[Jump to answer](#)

## Question 12

What is the output of the following code?

```
<?php
$list = new SplStack();

$list->push("a");
$list->push("b");
$list->push("c");

$list->rewind();

echo $list->pop();
```

- A. b
- B. Array
- C. Object
- D. c
- E. a

[Jump to answer](#)

## Question 13

What would be in `$matches[0]` after running the following code?

```
<?php
$pattern = '# \w*[aeiou]{2}\w*? #';
$string = "There's a moose loose about this house";
$matches = array();
preg_match_all($pattern, $string, $matches);
```

- A. "There's a moose loose about this house"
- B. An array containing " moose " and " about "
- C. "moose "
- D. An array containing " moose ", " loose ", " about " and " house "
- E. An array containing " moose " and " loose "

[Jump to answer](#)

## Question 14

What would be the output of this script?

```
<?php
ob_start();
echo "dreaming";
$ob = ob_get_contents();

echo strlen($ob);
ob_flush();
```

[textbox]

[Jump to answer](#)

## Question 15

What does status code 403 indicate?

- A. Forbidden
- B. Resource not found
- C. Not modified
- D. 403 is not a valid status code
- E. Moved permanently

[Jump to answer](#)

## Question 16

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);
```

- A. 9
- B. 7
- C. 5
- D. 3
- E. An error, because \$a is not an array

[Jump to answer](#)

## Question 17

What is the output of this line of code?

```
<?php
echo 8 + 0x8 + 80 + 0x80;
```

- A. 156
- B. 88
- C. 224
- D. 72
- E. 96

[Jump to answer](#)

## Question 18

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

- A. array\_merge()
- B. array\_combine()
- C. array\_splice()
- D. array\_keys()
- E. array\_intersect()

[Jump to answer](#)

## Question 19

What is the output of the following line of code?

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

- A. An error
- B. 18
- C. 117
- D. 115
- E. 14

[Jump to answer](#)

## Question 20

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;
```

- A. 1419
- B. 74
- C. 1519
- D. 1915
- E. 47

[Jump to answer](#)

## Question 21

What is the output of the following code?

```
<?php
$a = ["a" => "n",
      "o" => "t",
      "h" => "e"];

array_unshift($a, "f");

$a = extract($a);
echo $a;
```

- A. f
- B. 4
- C. Array
- D. 3
- E. n

[Jump to answer](#)

## Question 22

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

- A. url\_fopen\_allow
- B. url\_include\_allow
- C. allow\_open\_url
- D. allow\_url\_include
- E. include\_fopen\_dir
- F. open\_url\_dir
- G. open\_basedir

[Jump to answer](#)

## Question 23

Is the following valid PHP code?

```
<php>
echo 'There\'s a worm in my apple';
</php>}
```

- A. Yes
- B. No

[Jump to answer](#)

## Question 24

What is the output of:

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

- A. 105
- B. 100
- C. 050
- D. 005
- E. 150

[Jump to answer](#)

## Question 25

What would happen when the following code was run?

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

- A. A notice
- B. This tree is: oak (and a notice)
- C. An error
- D. This tree is: tree (and a notice)
- E. This tree is: (and a notice)

[Jump to answer](#)

## Question 26

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

- A. R7yu
- B. 9tur8UH4
- C. 8rutil
- D. AEN
- E. 111
- F. ablwesc\_99
- G. YU-8

[Jump to answer](#)

## Question 27

What is the output of the following code?

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

- A. 17
- B. 0
- C. 2
- D. 64
- E. 10

[Jump to answer](#)

## Question 28

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);
```

- A. 7-4-5-6-7-8-9
- B. 3-4-5-6-7-8-9-10-11
- C. 3-4-5-6-7-8-9
- D. 7-4-5-6-3-8-9
- E. 7-4-5-6-3-8-9-10-11

[Jump to answer](#)

## Question 29

What is the output of:

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

- A. 2110elephant
- B. 12
- C. 1210elephant
- D. 21
- E. 10elephant

[Jump to answer](#)

## Question 30

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;
```

[textbox]

[Jump to answer](#)

## Question 31

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));
```

- A. -!!&
- B. !&-&
- C. -&!
- D. !&-!
- E. -!&

[Jump to answer](#)

## Question 32

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

- A. \$\_FORMS
- B. \$\_COOKIE
- C. \$\_COOKIES
- D. \$\_REQUEST
- E. \$\_DATA
- F. \$\_ENVIRONMENT
- G. \$\_SERVER
- H. \$\_GLOBAL

[Jump to answer](#)

## Question 33

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

- A. Only one instance of the class can exist
- B. It implements ArrayAccess
- C. It is designed to store values of various types
- D. It uses static methods
- E. It takes responsibility for instantiating objects

[Jump to answer](#)

## Question 34

Is this statement true or false? “Methods declared as static must be called statically”

- A. True
- B. False

[Jump to answer](#)

## Question 35

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));
```

- A. Array ( [1] => Romeo and Juliet [4] => Bad Science [2] => To Kill A Mockingbird )
- B. Array ( [1] => Romeo and Juliet [2] => To Kill A Mockingbird [4] => Bad Science )
- C. 1
- D. Array ( [0] => Bad Science [1] => Romeo and Juliet [2] => To Kill A Mockingbird )
- E. 3

[Jump to answer](#)

## Question 36

ArrayAccess is an example of a:

- A. design pattern
- B. interface
- C. object
- D. class
- E. closure

[Jump to answer](#)



## Question 37

Using the notation `self::$property` refers to:

- A. A property of the current object
- B. The `$property` of the current object
- C. The class constant `$property` in this class
- D. The class property `$property` in this class
- E. A variable called `$property` in this class or any parent class.

[Jump to answer](#)

## Question 38

Which of the following is a valid namespace separator in PHP?

- A. \
- B. ::
- C. @
- D. /
- E. ->

[Jump to answer](#)

## Question 39

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();
```

- A. manygoodthingsmanygoodthings
- B. manygoodthings
- C. goodthingsgoodthings
- D. goodthingsmanygoodthings
- E. there is no output

[Jump to answer](#)

## Question 40

What is the output of the following code?

```
<?php
function values() {
    $j = 1;
    for($i=0; $i<4; $i++) {
        yield $j = $j + $i;
    }
}

foreach(values() as $v) {
    echo $v;
}
```

- A. 1234
- B. 1247
- C. 0123
- D. 1123
- E. Array

[Jump to answer](#)

## Question 41

What does the `html_errors` configuration directive do?

- A. Enable reporting of invalid HTML output
- B. Show PHP errors on webpages
- C. `html_errors` is not a valid configuration directive
- D. Apply HTML formatting to PHP errors
- E. Suppress browser errors

[Jump to answer](#)

## Question 42

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];
```

- A. "APPLES."
- B. "apples"
- C. "BREAD."
- D. nothing (no output)
- E. "fish"

[Jump to answer](#)

## Question 43

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?

- A. An warning because the interface isn't implemented
- B. A new CurrencyCheque object is created
- C. The method must be redefined in CurrencyCheque
- D. An error that the interface must be defined in the parent object
- E. A fatal error

[Jump to answer](#)

## Question 44

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");
```

- A. "world"
- B. an Exception
- C. nothing (no output)
- D. "hello"
- E. an error

[Jump to answer](#)

## Question 45

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;}
```

- A. using json\_decode()
- B. using the mcrypt extension
- C. using unserialize()
- D. using a database library
- E. you can't â€" this is a custom format

[Jump to answer](#)

## Question 46

Which of the following are true (choose three)?

- A. Anonymous functions can accept variables passed into them at call time
- B. Anonymous functions cannot be called recursively
- C. Closures are a kind of anonymous function
- D. Anonymous functions can have function names
- E. It is possible to return multiple values from an anonymous function
- F. Closures can have variable values “baked” in at declare time
- G. Anonymous functions are always created in the global scope

[Jump to answer](#)

## Question 47

What is the output of the following?

```
<?php  
echo chr((ord('a') + ord('A'))/2);
```

[textbox]

[Jump to answer](#)

## Question 48

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

- A. Use a custom stream wrapper
- B. file\_get\_contents() and explode()
- C. fopen() and fgetcsv()
- D. file() and explode()
- E. fopen(), fgets() and explode()

[Jump to answer](#)

## Question 49

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

- A. filter\_\* functions
- B. stripslashes()
- C. htmlspecialchars()
- D. database escape functions
- E. ctype\_\* functions
- F. strip\_tags()
- G. preg\_\* functions
- H. url\_decode()

[Jump to answer](#)

## Question 50

What is the output of the following?

```
<?php
$a = 0xf2 + 0x09;
$b = $a >> 3;
echo $b;
```

- A. 31
- B. 0
- C. 27
- D. 7
- E. 16

[Jump to answer](#)

## Question 51

What is the output of the following code?

```
<?php
trait A {
    public function b() {
        echo self::$name;
    }
}

class C {
    protected static $name="c";

    use A;
}

class D extends C {
    protected static $name="d";
}

$c = new D();
$c->b();
```

- A. d
- B. Error - method b() isn't available in class D
- C. Error - "use" must be the first thing in a class
- D. c
- E. Error - class property name isn't available in class D

[Jump to answer](#)

## Question 52

Which of the following would allow you to send a POST request to a remote resource via `file_get_contents()`?

- A. You can't, use the curl extension or an equivalent
- B. `stream_context(array("method" => "POST"));`
- C. `array("method"=>"post");`
- D. `stream_context_create("http_method" => "POST");`
- E. `stream_context_create(array("http" => array("method" => "POST")));`

[Jump to answer](#)

## Question 53

Which function would you use to re-order an array by its keys?

[textbox]

[Jump to answer](#)

## Question 54

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)

- A. \$c instanceof B
- B. \$d instanceof A
- C. \$d instanceof C
- D. \$e instanceof B
- E. \$e instanceof A
- F. \$c instanceof C
- G. \$a instanceof E

[Jump to answer](#)



## Question 55

The Active Record design pattern is used for which of the following?

- A. Creating audit trails of all file changes
- B. Easily storing and retrieving objects in the database
- C. Integrating PHP and Ruby on Rails projects
- D. Separating business and presentation logic within a system
- E. Managing output handlers

[Jump to answer](#)

## Question 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);
```

- A. an error
- B. Array ( [theme] => green )
- C. Array ( [theme] => green ) Array ( [theme] => green )
- D. Array ( [theme] => green ) Array ( [theme] => green ) Array ( [theme] => green )
- E. nothing

[Jump to answer](#)

## Question 57

What does the following code output?

```
<?php
$i = function ($j) {
    $i = $j + 4;
    $i++;
    return $i;
};
$j = 6;
echo $i($j);
```

- A. 10
- B. syntax error
- C. nothing
- D. 11
- E. 6

[Jump to answer](#)

## Question 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);
```

- A. 19
- B. 37
- C. 36
- D. 74
- E. nothing

[Jump to answer](#)

## Question 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);
```

- A. onenone
- B. noneonenoneonenone
- C. nonenoneonenone
- D. onenoneonenoneonenone
- E. noneonenone

[Jump to answer](#)

## Question 60

Which object method is automatically called when an object is cloned?

- A. \_\_copy()
- B. \_\_wakeup()
- C. \_\_drone()
- D. \_\_clone()
- E. \_\_call()

[Jump to answer](#)

## Question 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;
    }
}
```

- A. 0a1a2a3a
- B. 5a6a7a8a
- C. 0a0b0c0e
- D. 0a0b0c0e1a1b1c1e2a2b2c2e3a3b3c3e
- E. 5a5b5c5e6a6b6c6e7a7b7c7e8a8b8c8e

[Jump to answer](#)

## Question 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);
```

- A. This Sentence Contains Many Words
- B. This,contains,many,sentence,words
- C. This,Sentence,Contains,Many,Words
- D. This contains many sentence words
- E. This Contains Many Sentence Words

[Jump to answer](#)

## Question 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();
```

- A. `$s->setHandler("SoapFunctions");`
- B. `set_soap_class($s, "SoapFunctions");`
- C. `set_soap_handler($s, "SoapFunctions");`
- D. `$s->setClass("SoapFunctions");`
- E. nothing to add, the code above would work

[Jump to answer](#)

## Question 64

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

- A. `__walk()`
- B. `__sleep()`
- C. `__return()`
- D. `__call()`
- E. `__function()`
- F. `__add()`
- G. `__set()`

[Jump to answer](#)

## Question 65

Which function would transform the string “excellent PHP functions” into the string “Excellent PHP Functions”?

[textbox]

[Jump to answer](#)

## Question 66

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

- A. `mysqli_real_escape_string()`
- B. `mysqli_true_escape_string()`

- C. addslashes()
- D. using PDO and prepared statements
- E. htmlspecialchars()
- F. strip\_tags()

[Jump to answer](#)

## Question 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);
```

- A. Fatal error
- B. Catchable fatal error
- C. Mike
- D. November
- E. Syntax error

[Jump to answer](#)

## Question 68

What is the output of the following code?

```
<?php
$a = "Apple";
echo <<<'A'
pass me that $a
A;
```

- A. syntax error
- B. “pass me that Apple”
- C. “pass me that apple”
- D. fatal error
- E. pass me that \$a

[Jump to answer](#)

## Question 69

How would you change a SimpleXMLElement object into a DOMElement?

- A. convert SimpleXMLElement to XML, then create a DOMElement from the XML
- B. using simplexml\_export\_dom()
- C. using dom\_import\_simplexml()
- D. SimpleXMLElement::asDom()
- E. SimpleXMLElement::saveXML()

[Jump to answer](#)

## Question 70

What would you expect \$\_REQUEST to contain, with configuration settings as follows:

```
track_vars=1
request_order="GP"
variables_order="ESGPC"
```

- A. POST variables, overwritten by GET variables
- B. GET variables, overwritten by POST variables, COOKIE variables and SESSION variables, in that order
- C. COOKIE variables, overwritten by POST variables, GET variables, SESSION variables and ENV variables, in that order
- D. ENV variables, overwritten by SESSION variables, GET variables, POST variables and COOKIE variables, on that order
- E. GET variables, overwritten by POST variables

[Jump to answer](#)

# Sample Answers To Sample Questions

This section contains the answers and explanations of those answers

## Question 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())),'');
```

D. 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.

[Jump to question](#)

## Question 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

A trap! The function `construct()` isn't a constructor, that would need to be `__construct()` instead, so the function `M::m()` gets called twice.

[Jump to question](#)

## Question 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?

D. `$library->addChild("title", "Best book ever")`

None of the other options are valid or would work

[Jump to question](#)

## Question 4

What is the output of the following line of code:

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

E. 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  $4 \ll 3 = 32$

[Jump to question](#)

## Question 5

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

- A. An anonymous function
- D. An array containing an instantiated object as the first element, and the method name as the second element
- E. A string containing the function name

This question is about callbacks – only these three items in the list are valid ways to specify a callback <http://php.net/manual/en/language.types.callable.php>

[Jump to question](#)

## Question 6

How would you parse a web page at <http://example.com/page.php> as XML?

E. `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>

[Jump to question](#)

## Question 7

How many parameters does array\_merge() accept?

A. As many as you want

From php.net, the function prototype is `array array_merge ( array $array1 [, array $... ] )`

[Jump to question](#)

## Question 8

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

B. An array containing both associative and enumerated keys

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.

[Jump to question](#)

## Question 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](http://uk1.php.net/preg_match)). The pattern requires four lower case letters followed by a space, but doesn't anchor on to anything so "title " does match.

[Jump to question](#)

## Question 10

What is `PDO::query()` equivalent to?

E. `prepare()`, `execute()`, `fetch()`

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

[Jump to question](#)

## Question 11

Which HTTP status code asks a user to provide credentials?

C. 401

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

[Jump to question](#)

## Question 12

What is the output of the following code?

```
<?php
$list = new SplStack();

$list->push("a");
$list->push("b");
$list->push("c");

$list->rewind();

echo $list->pop();
```

D. c

The `SplStack` class extends `SplDoublyLinkedList`, but where the last thing in is the first thing out again. Rewinding this data structure has no effect, and so when we call `pop()` we get the last element that was added - in this case, “c”.

[Jump to question](#)

## Question 13

What would be in `$matches[0]` after running the following code?

```
<?php
$pattern = '# \w*[aeiou]{2}\w*? #';
$string = "There's a moose loose about this house";
$matches = array();
preg_match_all($pattern, $string, $matches);
```

B. An array containing " moose " and " about "

There are a few things to spot here – first of all, don’t be alarmed by all the punctuation in the regex, we’ll get to that. First the delimiters are `#` characters (you may know this as a pound or a hash, depending where your English is from). These are valid delimiters, the only requirement is that whatever you pick to put at the start must be matched at the end.

The delimiters have spaces inside them, so there must be spaces at the start and end of any matches. Then we must have any number of word characters – the ? after the \* means that the match isn't greedy (this happens again towards the end of the pattern) – followed by any two vowels, then any more word characters.

The first match is " moose " but this uses up the space before "loose", so that can't match, but " about " does, and then "house" which looks like it should match is missing the trailing space.

Still with me? `preg_match_all()` returns results in a nested array – the first element contains an array of all matches.

[Jump to question](#)

## Question 14

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.

[Jump to question](#)

## Question 15

What does status code 403 indicate?

A. Forbidden

Status code 403 means “Forbidden” – the user has provided credentials but still is not allowed to have access to this resource. See <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4> for more information.

[Jump to question](#)

## Question 16

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);
```

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.

[Jump to question](#)

## Question 17

What is the output of this line of code?

```
<?php
echo 8 + 0x8 + 80 + 0x80;
```

C. 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  $8+8+80+128 = 224$

[Jump to question](#)

## Question 18

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

A. array\_merge()

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

[Jump to question](#)

## Question 19

What is the output of the following line of code?

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

D. 115

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 octal is  $(8 + 1) = 9$ . Hopefully you knew that `ord('a')` was 97 and got 115 (`ord('A')` is 65, in case you're interested)

[Jump to question](#)

## Question 20

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;
```

E. 47

The function is never called.

[Jump to question](#)

## Question 21

What is the output of the following code?

```
<?php
$a = ["a" => "n",
      "o" => "t",
      "h" => "e"];

array_unshift($a, "f");

$a = extract($a);
echo $a;
```

D. 3

This example starts with the short array syntax to create an array, shifts a variable onto the front of the array, and then extracts it. Extract will ignore numerically-keyed variables, and create variables for any associative key/value pairs in the array. In this case, we assign the output of extract to \$a, which is the count of how many variables were extracted - 3, because the one we shifted on doesn't have an associative key.

[Jump to question](#)

## Question 22

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

D. allow\_url\_include  
G. open\_basedir

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

[Jump to question](#)

## Question 23

Is the following valid PHP code?

```
<php>
echo 'There\'s a worm in my apple';
</php>}
```

B. No

Those <php> style tags aren't valid for PHP.

[Jump to question](#)

## Question 24

What is the output of:

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

A. 105

`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.

[Jump to question](#)

## Question 25

What would happen when the following code was run?

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

D. This tree is: tree (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.

[Jump to question](#)

## Question 26

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

A. R7yu  
D. AEN  
F. ablwesc\_99

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

[Jump to question](#)

## Question 27

What is the output of the following code?

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

C. 2

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  $42 \& 22 \dots$  & 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.

[Jump to question](#)

## Question 28

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);
```

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

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 a `foreach`)

[Jump to question](#)

## Question 29

What is the output of:

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

A. 2110elephant

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.

[Jump to question](#)

## Question 30

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.

[Jump to question](#)

## Question 31

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));
```

E. -!&

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.

[Jump to question](#)

## Question 32

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

B. \$\_COOKIE  
D. \$\_REQUEST  
G. \$\_SERVER

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

[Jump to question](#)

## Question 33

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

A. Only one instance of the class can exist  
C. It is designed to store values of various types

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.

[Jump to question](#)

## Question 34

Is this statement true or false? “Methods declared as static must be called statically”

B. False

It's safe to call static methods dynamically, and you can do so. The opposite isn't true however; dynamic methods may not work when called statically so PHP will emit a PHP Strict Standards warning.

[Jump to question](#)

## Question 35

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));
```

C. 1

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

[Jump to question](#)

## Question 36

ArrayAccess is an example of a:

B. interface

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

[Jump to question](#)



## Question 37

Using the notation `self::$property` refers to:

D. The class property `$property` in this class

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

[Jump to question](#)

## Question 38

Which of the following is a valid namespace separator in PHP?

A. `\`

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

[Jump to question](#)

## Question 39

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();
```

D. goodthingsmanygoodthings

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 `$x` is 1. Look very closely at that `if` statement! There are no `{ }` brackets so only the “many” line is conditional, the “good” and “things” lines will run regardless (this is why indentation should follow program flow). On the second call, `$x` is 1 so the `if()` evaluates to true and we get the “many” line as well.

[Jump to question](#)

## Question 40

What is the output of the following code?

```
<?php  
function values() {  
    $j = 1;  
    for($i=0; $i<4; $i++) {  
        yield $j = $j + $i;  
    }  
}  
  
foreach(values() as $v) {  
    echo $v;  
}
```

B. 1247

Hopefully the generator syntax ( new in PHP 5.5 ) is familiar to you. The function `values()` is in fact a generator, because it contains `yield` statements instead of `return` statements. We can `foreach` over it, and each time we try to access a new value, the function gets called and will return us the next `yield` statement. Crucially, the generators retain their values in between calls, so both `$i` and `$j` keep their values as we go along.

[Jump to question](#)

## Question 41

What does the `html_errors` configuration directive do?

D. Apply HTML formatting to PHP errors

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

[Jump to question](#)

## Question 42

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];
```

D. nothing (no output)

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

[Jump to question](#)

## Question 43

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?

B. A new CurrencyCheque object is created

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.

[Jump to question](#)

## Question 44

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");
```

D. "hello"

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.

[Jump to question](#)

## Question 45

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;}
```

C. using `unserialize()`

This data is PHP's serialized format so you want 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.

[Jump to question](#)

## Question 46

Which of the following are true (choose three)?

- A. Anonymous functions can accept variables passed into them at call time
- C. Closures are a kind of anonymous function
- F. Closures can have variable values "baked" in at declare time

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>

[Jump to question](#)

## Question 47

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.

[Jump to question](#)

## Question 48

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

C. `fopen()` and `fgetcsv()`  
E. `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.

[Jump to question](#)

## Question 49

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

- A. filter\_\* functions
- E. ctype\_\* functions
- G. preg\_\* functions

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>

[Jump to question](#)

## Question 50

What is the output of the following?

```
<?php
$a = 0xf2 + 0x09;
$b = $a >> 3;
echo $b;
```

- A. 31

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. (alternatively, write down f2 and 9 in binary, and add them up in binary - most of us are safer adding up in decimal however)

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 1111 – which is 31 when you turn it back into decimal (hint: write from right to left above each digit: 1,2,4,8,16 .... and then add them up)

[Jump to question](#)

## Question 51

What is the output of the following code?

```
<?php
trait A {
    public function b() {
        echo self::$name;
    }
}

class C {
    protected static $name="c";

    use A;
}

class D extends C {
    protected static $name="d";
}

$c = new D();
$c->b();
```

D. c

The trait is declared correctly, and is applied to class C, at which point the `self` keyword will be resolved by PHP (this is why we have late static binding and the `static` keyword for getting the class names of descendants). Therefore in class D, we've inherited a method that specifically echoes `C::$name` and so it outputs "c";

[Jump to question](#)

## Question 52

Which of the following would allow you to send a POST request to a remote resource via `file_get_contents()`?

E. `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.

[Jump to question](#)



## Question 53

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>

[Jump to question](#)

## Question 54

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)

- B. \$d instanceof A
- C. \$d instanceof C
- E. \$e instanceof A
- F. \$c instanceof C

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.

[Jump to question](#)

## Question 55

The Active Record design pattern is used for which of the following?

B. Easily storing and retrieving objects in the database

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. More about this pattern: <http://www.martinfowler.com/eaCatalog/activeRecord.html>

[Jump to question](#)

## Question 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);
```

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

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).

[Jump to question](#)

## Question 57

What does the following code output?

```
<?php
$i = function ($j) {
    $i = $j + 4;
    $i++;
    return $i;
};
$j = 6;
echo $i($j);
```

D. 11

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.

[Jump to question](#)

## Question 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);
```

E. nothing

Nothing is returned by `swings()` so there's no output echoed (PHP functions return NULL by default). For added entertainment, `roundabout()` doesn't actually return anything either.

[Jump to question](#)

## Question 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);
```

B. noneonenoneonenone

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.

[Jump to question](#)

## Question 60

Which object method is automatically called when an object is cloned?

D. \_\_clone()

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>

[Jump to question](#)

## Question 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;
    }
}
```

A. 0a1a2a3a

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.

[Jump to question](#)

## Question 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);
```

B. This,contains,many,sentence,words

Applying `ucfirst()` to `$s` makes no difference, but `explode()`-ing 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.

[Jump to question](#)

## Question 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();
```

D. `$s->setClass("SoapFunctions");`

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>

[Jump to question](#)

## Question 64

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

B. `__sleep()`  
D. `__call()`  
G. `__set()`

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

[Jump to question](#)

## Question 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.

[Jump to question](#)

## Question 66

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

A. `mysqli_real_escape_string()`  
D. using PDO and prepared statements

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

[Jump to question](#)

## Question 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);
```

C. Mike

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` instead of using `N`. Late static binding means we could use `static` instead of `self` and get "November" as the output.

[Jump to question](#)

## Question 68

What is the output of the following code?



```
<?php
$a = "Apple";
echo <<<'A'
pass me that $a
A;
```

E. 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.syntax.nowdoc>

[Jump to question](#)

## Question 69

How would you change a SimpleXMLElement object into a DOMElement?

C. using dom\_import\_simplexml()

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 DOMElement 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>

[Jump to question](#)

## Question 70

What would you expect \$\_REQUEST to contain, with configuration settings as follows:

```
track_vars=1
request_order="GP"
variables_order="ESGPC"
```

E. GET variables, overwritten by POST variables

Don't be distracted by the other settings - the only one that matters here is `request_order`. `track_vars` needs to be on, without it the superglobals aren't created at all (but it's been on by default since 4.0.3!). If `request_order` wasn't set, we'd fall back to `variables_order`, but in this case it is set, and the value "GP" means that first the GET variables will be placed into the `$_REQUEST` array, then the POST variables will be added and may overwrite anything that was already there.

[Jump to question](#)