# Zend Certified PHP Engineer (ZCPE) Practice Test (Sample)

## Study Guide

BY EXAMZIFY

**Everything you need from our exam experts!**

**Notice: Examzify makes every reasonable effort to obtain from reliable sources accurate, complete, and timely information about this product.**

# Table of Contents

# Introduction

**Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn’t just about memorization, it’s about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.**

**This guide was designed to help you do exactly that.**

**Whether you’re preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You’ll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it’s correct.**

**The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It’s ideal for learners who want to:**

- **Practice answering questions under realistic conditions,**
- **Improve accuracy and speed,**
- **Review explanations to strengthen weak areas, and**
- **Approach the exam with greater confidence.**

**We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.**

**Remember: successful test preparation isn’t about getting every question right the first time, it’s about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.**

**Let’s begin.**

# How to Use This Guide

**This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:**

## 1. Start with a Diagnostic Review

**Skim through the questions to get a sense of what you know and what you need to focus on. Don't worry about getting everything right, your goal is to identify knowledge gaps early.**

## 2. Study in Short, Focused Sessions

**Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations, and take breaks to retain information better.**

## 3. Learn from the Explanations

**After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.**

## 4. Track Your Progress

**Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.**

## 5. Simulate the Real Exam

**Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.**

## 6. Repeat and Review

**Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning.**

## 7. Use Other Tools

**Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.**

**There’s no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly — adapt the tips above to fit your pace and learning style. You've got this!**

# Questions

**1. What does XML stand for?**

**A. eXtensible Markup Language**

**B. Exponential Markup Language**

**C. Extended Markup Language**

**D. Executable Markup Language**

**2. What is a trait in PHP?**

**A. A built-in PHP function**

**B. A mechanism for code reuse**

**C. An object-oriented feature**

**D. A type of variable**

**3. What is a closure in PHP?**

**A. A function that is defined within a class**

**B. An anonymous function that can capture variables from its surrounding scope**

**C. A reserved block of code that does not require a return statement**

**D. A function that can only be called once**

**4. What is the purpose of the preg_match() function in PHP?**

**A. To perform calculations**

**B. To match a string against a regular expression**

**C. To escape special characters in a string**

**D. To format dates**

**5. What is the significance of the array NOT IDENTICAL operator ( !== ) in PHP?**

**A. It checks for different memory references between two arrays**

**B. It checks if arrays look the same but may differ in type**

**C. It is used to confirm the elements in arrays are different**

**D. It confirms that two arrays are not identical by both structure and type**

**6. Why is it important to escape incoming data in PHP?**

**A. To improve load times**

**B. To prevent SQL injection attacks**

**C. To enhance code readability**

**D. To ensure data integrity**

**7. How do you remove an element from an array in PHP?**

**A. Using the delete() function**

**B. Using the remove() function**

**C. Using the escape() function**

**D. Using the unset() function**

**8. What is wrong with the provided XML document?**

**A. It has a missing closing tag**

**B. It contains an invalid character**

**C. Nothing**

**D. It is missing the XML declaration**

**9. How can you declare a PHP class property?**

**A. By defining its type**

**B. By specifying its visibility followed by the variable name**

**C. By initializing it within a method**

**D. By declaring it within the constructor**

**10. What does the operator '==' do when comparing two arrays in PHP?**

**A. It checks if the arrays are strictly equal**

**B. It checks if both arrays have identical types**

**C. It checks if the arrays have the same key/value pairs**

**D. It merges the two arrays together**

# Answers

1. **A**
2. **B**
3. **B**
4. **B**
5. **D**
6. **B**
7. **D**
8. **C**
9. **B**
10. **C**

# Explanations

### 1. What does XML stand for?

A. eXtensible Markup Language

B. Exponential Markup Language

C. Extended Markup Language

D. Executable Markup Language

**XML stands for eXtensible Markup Language. It's a versatile markup language that is designed to store and transport data in a format that is both human-readable and machine-readable. The term "eXtensible" indicates that XML allows for the creation of custom tags, thereby providing a flexible way to describe any kind of data structure. XML is widely used for various purposes, such as web services, data storage, and configuration files. Its ability to define complex hierarchies and structures makes it particularly suitable for representing structured data and interchanging information across different systems and platforms. This extensibility is a fundamental characteristic that distinguishes XML from other markup languages, allowing developers to tailor the markup to their specific needs and applications.**

### 2. What is a trait in PHP?

A. A built-in PHP function

B. A mechanism for code reuse

C. An object-oriented feature

D. A type of variable

**A trait in PHP is specifically designed as a mechanism for code reuse. Traits allow developers to create reusable components that can be included in multiple classes. This feature is particularly useful in PHP, where multiple inheritance is not supported. By using traits, developers can include methods in a class without having to share a common ancestor, fostering a more modular and organized code structure. Traits help to reduce code duplication and increase maintainability, as they allow for shared functionality to be defined once and used across multiple classes. When a class uses a trait, it can access the methods provided by the trait, enabling a flexible mix of behaviors. While it is true that traits are often associated with object-oriented programming, referring to them merely as an object-oriented feature does not capture their main purpose of promoting code reuse. Similarly, traits are not built-in functions nor a type of variable, but rather a mechanism that empowers developers to effectively structure their applications.**

## 3. What is a closure in PHP?

A. A function that is defined within a class

B. An anonymous function that can capture variables from its surrounding scope

C. A reserved block of code that does not require a return statement

D. A function that can only be called once

**A closure in PHP refers specifically to an anonymous function that can capture and use variables from its surrounding scope. This means that it has the ability to access variables that are defined outside of its own body. This feature is particularly useful for scenarios where you need a function to maintain some state or configuration without having to use global variables or other forms of state management. Closures are often used in functional programming styles in PHP, especially when combined with functions that require a callback, such as array mapping or filtering methods. By allowing the function to capture variables from its environment, closures provide a more flexible, concise, and expressive way to handle functions in dynamic contexts. Understanding closures is essential, as they are a powerful feature of PHP that enhance coding efficiency and maintainability.**

## 4. What is the purpose of the preg_match() function in PHP?

A. To perform calculations

B. To match a string against a regular expression

C. To escape special characters in a string

D. To format dates

**The preg_match() function in PHP is designed specifically to match a given string against a specified regular expression pattern. This function evaluates whether the string adheres to the defined rules articulated in the regex, returning true if there is a match and false if there is not. Regular expressions are powerful tools used for pattern matching within strings, allowing developers to search, validate, or manipulate strings based on complex criteria. The preg_match() function returns additional information about the match if desired, making it versatile for various string processing tasks, such as validating user input or parsing text. In this context, the other options do not relate to the functionality of preg_match(). The function is not intended for performing calculations, escaping special characters, or formatting dates, as these tasks are accomplished by different PHP functions tailored to those specific purposes.**

## 5. What is the significance of the array NOT IDENTICAL operator ( !== ) in PHP?

A. It checks for different memory references between two arrays

B. It checks if arrays look the same but may differ in type

C. It is used to confirm the elements in arrays are different

D. It confirms that two arrays are not identical by both structure and type

The array NOT IDENTICAL operator (`!==`) in PHP is significant because it provides a strict comparison between two arrays. This means it checks not only the values contained in the arrays but also their data types and structure. When using this operator, if two arrays are not the same in terms of both the arrangement of elements and the types of those elements, the result will be true.  For example, if one array contains integer values and the other contains string representations of those values, they will not be considered identical due to the difference in types. Similarly, if two arrays have the same values but in a different order or key structure, they will also return true when compared with the NOT IDENTICAL operator. This makes it a valuable tool for ensuring that two arrays are truly distinct in all defined aspects, providing a more thorough comparison than merely checking the values alone.

## 6. Why is it important to escape incoming data in PHP?

A. To improve load times

B. To prevent SQL injection attacks

C. To enhance code readability

D. To ensure data integrity

Escaping incoming data in PHP is crucial for preventing SQL injection attacks, which are one of the most common security vulnerabilities in web applications. When user inputs are directly included in SQL queries without proper validation or escaping, malicious users can manipulate the input to execute arbitrary SQL code. This can lead to unauthorized access to the database, data leakage, data destruction, or even full control over the database server.  By escaping incoming data, developers mitigate this risk by sanitizing user input, thus ensuring that any special characters are treated as plain text rather than executable code. This practice effectively neutralizes the threat posed by malicious input.  While improving load times, enhancing code readability, or ensuring data integrity are all important aspects of coding and application design, they are not the primary reason for escaping incoming data in the context of web security. The fundamental goal of this practice is to secure the application against SQL injection, making option B the key focus for developers concerned with maintaining secure PHP applications.

## 7. How do you remove an element from an array in PHP?

- A. Using the delete() function
- B. Using the remove() function
- C. Using the escape() function
- D. Using the unset() function

When it comes to removing an element from an array in PHP, the unset() function is the definitive choice because it directly modifies the array by removing a specified element. When you apply unset() to an element in an array, it effectively unsets or deletes that specific key and its associated value from the array. This function takes as an argument the variable or array element you wish to remove and does not return a value; instead, it changes the structure of the original array. For instance, if you have an array defined as $array = ['apple', 'banana', 'cherry'], executing unset($array[1]) would result in $array containing just 'apple' and 'cherry', effectively removing 'banana'. The other choices presented do not correspond to actual PHP functions that would accomplish the task of removing an element from an array. This makes unset() the only valid and functional option for the task. Thus, it is crucial to understand the role and functioning of unset() within PHP to effectively manipulate arrays.

## 8. What is wrong with the provided XML document?

- A. It has a missing closing tag
- B. It contains an invalid character
- C. Nothing
- D. It is missing the XML declaration

The assertion that "Nothing" is wrong with the provided XML document indicates that the XML is well-formed, meaning it adheres to the basic rules of XML structure. A well-formed XML document must have a single root element, properly nested tags, and all tags must be closed correctly. If an XML document has no missing closing tags, does not contain any invalid characters (such as unescaped ampersands or other reserved symbols that are not properly represented), and includes the essential elements of a valid XML structure, including optional documentation formatting like the XML declaration, then it can be accurately considered as well-formed. The correctness of this answer suggests that the document has been validated and passes all structural checks defined by XML standards. This could also mean that the XML declaration (e.g., `<?xml version="1.0"?>`), while helpful for indicating the version of XML being used, is not a strict requirement for the document to be deemed well-formed. Therefore, the absence of the XML declaration does not, in itself, render the document invalid, leaving the notion that if no structural issues exist, the document is correctly formatted.

## 9. How can you declare a PHP class property?

A. By defining its type

B. By specifying its visibility followed by the variable name

C. By initializing it within a method

D. By declaring it within the constructor

**To declare a PHP class property, you specify its visibility (such as public, protected, or private) followed by the variable name. This approach allows you to define the scope and accessibility of the property within the class. For example: ```php class Example { public $propertyName; } ``` In this declaration, "public" indicates that any code can access this property, while "propertyName" is the name of the variable within that class. This standard structure is essential for properly setting up properties in PHP objects, ensuring they are appropriately encapsulated according to object-oriented principles. The other methods mentioned in the options refer to different aspects of working with class properties. Defining a type is part of type hinting or type declarations, which is different from the property declaration itself. Initializing a property within a method or declaring it within the constructor are not considered direct declarations of the property; rather, these are actions taken to assign a value or create an instance of the property after it has already been declared.**

## 10. What does the operator '==' do when comparing two arrays in PHP?

A. It checks if the arrays are strictly equal

B. It checks if both arrays have identical types

C. It checks if the arrays have the same key/value pairs

D. It merges the two arrays together

**The operator '==' in PHP performs a comparison of two arrays to determine if they have the same key/value pairs, regardless of the order in which those pairs are arranged. This means that if two arrays contain exactly the same keys associated with the same values, they will be considered equal, even if the keys are in a different order. For example, if you have one array with the structure `['a' => 1, 'b' => 2]` and another array structured as `['b' => 2, 'a' => 1]`, using the '==' operator will return true, indicating that these arrays are equal based solely on their content. In contrast, the other options describe different behaviors that do not accurately reflect the functionality of the '==' operator. Specifically, strictly equal comparisons or type checks are not part of the functionality of '==', which focuses on value equality rather than strict type identity or merging functionality.**

# Next Steps

**Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.**

**As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.**

**If you need help, have suggestions, or want to share feedback, we’d love to hear from you. Reach out to our team at hello@examzify.com.**

**Or visit your dedicated course page for more study tools and resources:**

**https://zendcertifiedphpengineer.examzify.com**

**We wish you the very best on your exam journey. You've got this!**