# Everlytic Developer Assessment

|  |  |
| --- | --- |
| Name | ADEDOJA ADEDAMOLA |

|  |  |
| --- | --- |
| Date | 09/03/2018 |

Please complete the answers to the questions below. The assessment should take roughly 30 minutes.

### What is the difference between public, protected and private in a class definition?

|  |
| --- |
| Public: It allows the properties or methods be accessed from everywhere in the program. |
| Protected: It allows the properties or methods be accessed only from inside the class in which they are declared or any other class that extends them. |
| Private: It allows the properties or methods be accessed only within its own class. |

### Given this code: function doSomething(&$foo) { $bar = $foo; $foo += 1; return $foo; } $value = 3; $result = doSomething($value); echo "value: $value, result: $result"; What will be output to screen and why?

|  |
| --- |
| value: 4, result: 4 |
| This is because the ‘&’ operator used means the variable ‘$foo’ is passed by reference |
| instead of value, so any changes made to the ‘$foo’ was made to ‘$value’ too |

### What is wrong with this query: "SELECT \* FROM table WHERE id = $\_POST[ 'id' ]"?

|  |
| --- |
| Using ‘\*’ selects all the columns in the table, for a table with a significant amount of data, |
| that will be a lot of data, and this cause performance issues. Most of the time, not all of the |
| columns will be needed, it is better to just select the columns needed. |
| Also, in the case of ‘JOIN’ query, this might cause conflicts in the results |

### What is the cause of this warning: 'Warning: Cannot modify header information - headers already sent', and what is a good practice to prevent it?

|  |
| --- |
| Sometimes, when attempting to modify the header information after output has been sent |
| to the client. Also, the warning is usually caused because of white spaces. White spaces |
| that sometimes appear before the opening ‘<?php ’ tag or after the closing tag. A good way |
| to prevent this is to make sure there no spaces before the opening php tag and to omit the |
| Closing php tag ‘?>’ from files containing only PHP |

### What is wrong with this code: class Foo { protected $bar; public function \_\_construct() { $this->bar = 1; } public static function doSomething() { return $this->bar; } }

|  |
| --- |
| The use of ‘$this’ in the static function/method. Static functions/methods can be called |
| without an instance of the object created. ‘$this’ is not available inside methods declared |
| as static. Also, static properties cannot be accessed through the object using the arrow |
| operator (-> ), they are accessed using ‘::’ |

### Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

<?php

for ($i = 1; $i <= 100; $i++) {

$isMultipleOfThree = (0 === $i % 3);

$isMultipleOfFive = (0 === $i % 5);

if (!$isMultipleOfThree && !$isMultipleOfFive) {

echo $i . "<br />";

continue;

}

if ($isMultipleOfThree) {

echo 'Fizz';

}

if ($isMultipleOfFive) {

echo 'Buzz';

}

echo "<br />";

}

### **What does the following code do? Explain what’s going on there.**

$date = '08/26/2003';

print preg\_replace('/([0-9]+)\/([0-9]+)\/([0-9]+)/'‚ '$2/$1/$3', $date);  
preg\_replace is a regular expression search and replace function. It’s

|  |
| --- |
| Arguments are preg\_replace($pattern, $replacement, $subject). In this case, the pattern |
| is to divide the date string into 3 capturing groups, with each of them having characters |
| matching between 0 and 9, making sure no alphabets gets. The replacement is a |
| Rearrangement, with the first group becoming second and the second becoming first i.e |
| ‘$1/$2/$3’ becomes ‘$2/$1/$3’. The example above prints out ‘26/08/2003’ |

### **Given a line of text $string, how would you write a regular expression to strip all the HTML tags from it?** Using a preg\_replace, the pattern for the regular expression would be

|  |
| --- |
| /<(.\*?)(\h\*).\*?>(.\*?)<\/\1>/ |
| This looks for html tags, any string with characters starting with a ‘<’ and ends with ‘>’ or |
| ‘/>’. Also, it is any character (.\*?). Also checks for white spaces |

### A palindrome is a word that reads the same backward or forward. Write a function that checks is a given word is a palindrome. Characters case should be ignored. EG. Deleveled is a palindrome and should return true as character case is ignored.

### <?php class Palindrome

### {

### public static function isPalindrome($word)

{

}

### } echo Palindrome::isPalindrome('Deleveled');

<?php

class Palindrome

{

public static function isPalindrome($word)

{

$wordToLower = strtolower($word);

if (strrev($wordToLower) == $wordToLower) {

return 'true';

} else {

return 'false';

}

}

}

echo Palindrome::isPalindrome('Deleveled');

### What security issue is prevalent in the code below and how would you fix it? <?php $messageStmt = $db->query('select message\_text from messages where message\_id = 1'); $messageStmt->execute(); $message = $messageStmt->fetch(PDO::FETCH\_OBJ); ?> <div><?= $message->message\_text; ?></div> SQL Injection is the security issue, Can be fixed using PDO prepared statements.

### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Write an inner join for the following tables

|  |
| --- |
| SELECT User.UserKey, User.FirstName, User.LastName, User.Email, Address.Address1 |
| FROM User |
| INNER JOIN Address ON User.UserKey = Address.UsrKey |

### 