# Everlytic Developer Assessment

|  |  |
| --- | --- |
| Name | Keith Msimango |

|  |  |
| --- | --- |
| Date | 20023/07/16 |

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 – are members / function accessible from out side this class |
| Protected – are not accessible form out side the class but can be access through inherited classes |
| Private -> these function or members are sickly only accessible within the same |

### 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 |
| Valuating line $result = doSomething($value); |
| we tell the compiler to execute the function doSomething and passing the value 3, we assign bar variable to 3 as variable foo is equal to 3 at this point in time, the nest line simply adds a new the assigned value 1 to foo which is 4 and the current result now is 4. Should we run the code function doSomgthing will result into 5 rather then 4, due to we call the reference of $foo and its last iteration was changed to 4 |

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

|  |
| --- |
| We can use $\_POST[‘id’] directly in and an sql statement without using quotation in this string should be at least is should be *“ . $\_POST[‘id’] . ”* to isolation the string from php code |
| The word table is a designated reserved word, and should there be a table in the database called table we can use backticks around the word to make this statement executable. |
|  |

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

|  |
| --- |
| Usually this is meansn something was sent to the browser before the header function was called. |
| Make sure there are no empty spaces before <?php at the beginning of the file |
|  |

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

|  |
| --- |
| We can user $this in a static function |
| Using '$this' when not in object foo |
|  |

### 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".

public static function fizzBuz()

{

for ($number = 1; $numbers < 101; $number++) {

$display = “”;

if ($number % 3 === 0 && $number % 5 === 0) {

$display = “FizzBuzz”;

} else if ($number % 3 === 0) {

$display = “Fizz”;

} else if ($number % 5 === 0) {

$display = ‘Buzz’;

} else {

display = $number;

}

echo $display . “<br>”;

}

}

self::fizzBuz();

### **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);  
date -> date format display

|  |
| --- |
| “/../” start and ends the regulare expression  We perform a regular expression search and replace |
| ([0-9] match numbers form 0 – 9  \/ Escape back slash  ([0-9]+ match numbers form 0 – 9 include characters ot its left 1 or more  \/ Escape back slash  ([0-9]+ match numbers form 0 – 9 include characters ot its left 1 or more  This is $date is then transformed from its original '08/26/2003' to 26/08/2003  '$2/$1/$3', means, the string is matched into 4 sections / array and we re-arranged the sequence of the string  $0 = 08/26/2003  $1 = 08  $2 = 26  $3 = 2003  And that’s has been replaced with the new array parameters  With the new string $2/$1/$3 giving us  26/08/2003 |

### **Given a line of text $string, how would you write a regular expression to strip all the HTML tags from it?**

|  |
| --- |
| I would user Strip\_tags() |
| $string = preg\_replace(‘/<[^>]\*>/’, ‘’, $string) |
| echo string; |
|  |

### 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)

{

$reverse\_string = strrev($word);

if ($reverse\_string === $word) {

return TRUE;

} else {

return NULL;

}

}

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

### Considering message\_text stores a combination of html and text. 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><?php echo $message->message\_text; ?></div> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

$this->\_results = $this->\_query->fetchAll(PDO::FETCH\_OBJ);

I would use the mysqli\_real\_escapes\_string().

<?php  
$messageStmt = $db->query(‘select ‘. mysqli\_real\_escape\_string(‘message\_text’) . ‘ from `messages` where `message\_id` = “1” ‘);  
$messageStmt->execute();  
$message = $messageStmt->fetchAll(PDO::FETCH\_OBJ);  
?>  
<div><?php echo $message->message\_text; ?></div>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Write an inner join for the following tables

|  |
| --- |
| SELECT u.UserKey, u.Firstname, u.LastName, u.Email, addr.Address1, addr.Address2, addr.City, addr.State, addr.ZipCode  FROM User u  INNER JOIN address .addr on (u.Userkey = addr.UsKey) |
|  |

### Complete the JS function below that validates the conditions of a password: 1. The password must be greater then 7 characters 2. The first character must be a capital letter 3. The password must contain at least one number

function isPasswordValid(password) {

var isValid = false;

// Write you logic here

if (password.length > 7) {

if (password[0] === password[0].toUpperCase()){

if (hasNumber(password)) {

isValid = true;

} else {

isValid = ‘the password must contain at least one number’;

}

} else {

isValid = ‘the first characters must be a capital letter’;

}

} else {

isValid = ‘Your password must be 8 characters long of more’;

}

return isValid;

}

function hasNumber(string) {

return /\d/.test(string);

}

let string = ‘your string here’;

console.log(isPasswordValid(string))

**Side note, if this is a js function why is there a ‘$’ in the function declaration?**