Everlytic Developer Assessment

Name	Lee Aaron
Date	14/05/2022

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

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

Public visibility means that any method/variable defined in the class and declared as public, is visible and available to access, not only to child classes, but to any object instantiated from the class

Protected visibility means that any methods/variables that are defined as protected in the parent class, are only visible and allow access, to child classes (classes extending the parent class)

Private visibility means that any methods/variables that are defined as a private in the parent class, are only visible and accessible within the parent class itself, and nowhere else.

2. 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?

The output will be value: 4, result: 4,

This is because the doSomething method is using the $foo parameter as a reference, this means that it points to the same memory slot allocated to the passed in argument ($value),

So when $foo is altered in the doSomething method, it also changes the value of $value, as they are pointing to same allocation in memory, and therefore when $foo is altered (3 + 1 = 4), $value also changes and is equal to 4.

but because the method also returns $foo as the value, when doSomething is called, it returns the altered $foo variable (3 + 1 = 4), and $result is assigned the returned value, and $result also equals 4.

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

    1. This query is taking the value directly from the input form, and placing it directly in the query without escaping, validating or cleaning the data up, which means that this allows the possibility for SQL injection, meaning that a hacker can enter malicious SQL code in the input and that could be executed in the database when this query is run, this can be fixed by escaping and cleaning the input and/or by running it through a prepared query

    2. This query could also fail because if id field is a data type such a UUID, using a RDBMs such as Microsoft SQL Server, and you pass in a value as a basic string or integer, the query will throw an exception as the data type don't match, this can be fixed by validating that the input is the correct data-type

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

There can be multiple causes, such as

    1. whitespaces before or after the PHP closing and opening tags, this can be avoiding by making sure the code is neat, tidy and formatted correctly

    2. If there is an echo, print or dump in the code, that comes before the php headers are sent, this can be avoided by making sure there are no un-needed dumps, prints or echos in the code and moving any HTML blocks to after the headers get sent

5. What is wrong with this code:

```
class Foo
{
    protected $bar;

    public function __construct()
    {
         $this->bar = 1;
    }

    public static function doSomething()
    {
        return $this->bar;
    }
}
```

$this->bar is not accessible in the doSomething method, as $this is looking at the scope of the class in an instantiation of this class, but because the method doSomething is a static function, it means that it can be called without instantiating a class, and therefore the class variable $bar would not exist when the method is called.

6. 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".

```
for ($i = 0; $i <= 100; $i++) {
    switch ($i) {
        case (($i % 3 === 0) && ($i % 5 === 0)):
            //Using nl2br for neater line by line displaying
            echo nl2br("FizzBuzz\n\r");
            break;

        case ($i % 3 === 0):
            //Using nl2br for neater line by line displaying
            echo nl2br("Fizz\n\r");
            break;

        case ($i % 5 === 0):
            //Using nl2br for neater line by line displaying
            echo nl2br("Buzz\n\r");
            break;

        default:
            //Using nl2br for neater line by line displaying
            echo nl2br(sprintf("%s\n\r", $i));
            break;
    }
}
```

7. 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);
```

This code is using regex to look through the date variable for the format matching

1. / = (start of data string)
2. ([0-9]+) = any digit between 0 and 9 showing up at least once (can have unlimited digits)
3. \/ = a string literal version of a forward slash ('/')
4. Copies that pattern of 2 and 3 (2 more times)
5. / = (end of data string)

This when it finds a match to the pattern in the string, it then replaces it with the string '$2/$1/$3'

---

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

/(\<)([\/]{0,1})[\w\s!@#$%^&*()_+-=\[\]\{\}\|\\:;"',.?\/]+(\>)/

---

1. This would look for a less than '<' to indicate the start of a tag at the start of the string
2. And then it would look for 0 or 1 ending slashes '/' to include opening and closing tags too
3. Then look for letter (case insensitive) or number or string literal special character unlimited times to include attributes
4. Then look for a greater than '>' to indicate the end of the tag at the end of the string

---

---

9. 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)
   {
     $lowerWord = strtolower($word);
     $reversedLoweredWord = strrev($lowerWord);
```

```
    Return $lowerWord === $reversedLoweredWord;
  }
}

echo Palindrome::isPalindrome('Deleveled');
```

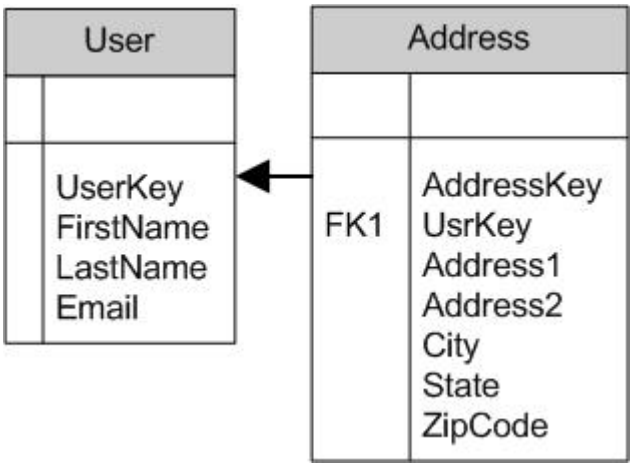
10. Considering message_text stores a combination of html and text. What security issue is prevalent in the code below and how would you fix it?

```
<?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>
```

Because this data is retrieved directly from the database and is being inserted directly into the html, this is creating an opening to cross-site scripting, which is a form of attack that allows an attacker to insert malicious code (like a JS script) into the html page, allowing the attacker to do things like stealing people's sessions or messing with sites page

This could be fixed by escaping, validating and cleaning the message text string in PHP, such as stripping html tags from the string, and making sure that that any html specific characters are made literal, so that they are read as actual text and not html characters (many templating engines have these feature included)

11. Write an inner join for the following tables



SELECT * FROM Address a INNER JOIN User u ON u.UserKey = a.UsrKey

12. 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 &&
          $password[0].toUpperCase() === $password[0] &&
          /\d/.test($password)
      ) {
            isValid = true;
      }

    return isValid;
}
```