

Abdel Mahmoud

Professor Kholodovsky

CSC 226

10/05/24

Assignment 2

1. **What does it mean for PHP to be an interpreted language? How does this differ from a compiled language?**
 - A. With php being an interpreted language, everything is ran line by line by the interpreter every time you want to test or run the code. This differs from a compiled language in the sense that when the code is run, it is compiled into the machine code all at once. The interpretation that comes from php makes it arguably more efficient since you are able to run everything line by line.
2. **Discuss the advantages and disadvantages of using an interpreted language like PHP over a compiled language.**
 - A. The obvious disadvantages when using an interpreted language like PHP over a compiled language would be when the program is run. It has to be interpreted and put into machine code in order to run, which may cause it to run slower than its counterpart. On the contrary, PHP is capable of being advantageous when it comes to cross platforms, also allowing for easier development through the use of dynamic typing and accessible debugging to better improve your current projects.

PHP Arrays

3. **Explain the difference between indexed arrays and associative arrays in PHP. Provide examples of both.**
 - A. The difference between indexed arrays and associative rays in PHP is that indexed arrays use numbers to allow for the referencing of data in an array. An example of this would be creating positions from 0 - 3 for different colors or shapes; the corresponding position would link you to that particular color/shape. An associative array however, allows you to access data via keys that are essentially

logged when there is a received input data into the array. This is best represented when searching in an online store for an electronic device. You would press on the string and it would lead you to particular results.

4. **How can you iterate through an associative array in PHP? Provide code demonstrating at least two different methods.**

A. When iterating through an associative array in PHP you can use a foreach loop or a loop that uses key. #Assume all necessary declared variables.

```
Foreach: foreach ($array as $i => $value) {  
#assume "meat of coding"  
}  
?>
```

```
Using key: $keyed_array = array_keys($array)  
for ($i = 0; count(array_keys($array)); ++$i)  
{ echo $array[$keys[$i]];  
}  
?>
```

5. **What will the following code output, and why?**

```
$arr = [1 => 'apple', 2 => 'banana', 'key' => 'orange'];  
echo $arr[1];  
echo $arr['key'];
```

A. The following code output for “\$arr = [1 => 'apple', 2 => 'banana', 'key' => 'orange']; echo \$arr[1]; echo \$arr['key'];” would print “appleorange” with banana being left out due to the value being associated with it (2) not being referenced. Additionally, due to the lack of space or line breaking that is not provided, apple and orange are placed right after each other.

PHP Classes

6. **Define a class in PHP that represents a Car with properties like make, model, and year. Include methods for starting the car and stopping it.**

A. <?php

```

class Car
{ public $make;
public $model;
public $year;
public function __construct($make, $model, $year)
{ $this->make = $make; $this->model = $model; $this->year = $year; }
public function start()
{ echo "The $this->year $this->make $this->model is starting up...\n"; }
public function stop() {
echo "The $this->year $this->make $this->model is stopping...\n";
}
}

```

The echo statements essentially just say that the particular car (whatever it may be in the context) is starting, and stopping

7. **What is the purpose of the `__construct()` method in PHP classes? How is it different from regular methods?**

- A. The purpose of the construct method is to provide a constructor for a class where it gives the freedom to the individual to develop new objects with their associated variables, or even change the variables with little hassle. This differs from regular methods in the sense that the method will always be called when the object is set within the code, meaning it can not be used as a normal function, only object creation.

PHP Variables

8. **What are the different types of variables in PHP?**

- A. There are many types of variables in PHP. They include: integers, arrays, objects, strings, floats, null, booleans, and resources.

9. **What does the `$this` keyword refer to in PHP? Provide an example of how it is used within a class.**

- A. The "`$this`" keyword refers to a reference to the object's variable or instance that is currently highlighted within a class method. An example of this would be

```

<?php
class Fruit
{ public $color;
public $fruit;
}

```

```

public function start() {
    echo "Your favorite fruit is a " . $this->color . " " . $this->fruit . "\n"; }
}
?>

```

/* As we can see, the “\$this” is essentially picking or pinpointing the particular object we are working on right now. In this context of this code, the output would say something along the lines of “Your favorite fruit is a red apple.”
*/

PHP If Statements

10. **Write an if-else statement that checks whether a number is positive, negative, or zero, and prints the appropriate message.**

A.

```

# assume all necessary variables.
if ($number > 0
    { echo "This is a positive number.\n";
  } elseif ($number < 0)
    { echo "This is a negative number.\n"; }
else { echo "This number is exactly zero.\n";
}

```

11. **Explain the difference between == and === in PHP if statements. Provide an example where they would behave differently.**

A.

"==" Only compares values and the "===" compares both the value and its type. An example of this would be

```

if (8 == "8") {

}

```

This would return true since the values are the same.

```

if (8 === "8") {

}

```

this returns false because it is no longer 5 with 5, but rather 5 with a string.

12. What is a ternary operator in PHP? Rewrite the following if-else statement using the ternary operator:

Copy code

```
if ($a > $b) {  
    echo "a is greater";  
} else {  
    echo "b is greater";  
}
```

A. A ternary operator in PHP essentially allows for another way of writing an if-else statement. In this particular way, if-else statements are often considered to be “shorter” but still allow for the same end results.

```
if ($a > $b) {  
    echo "a is greater";  
} else {  
    echo "b is greater";  
}
```

code before the ternary operator is introduced.

```
echo ($a > $b) ? "a" : "b";
```

code afterwards is expressed through the "?" is highlighted if a is greater than b, the ":" represents the false condition meaning b is greater

13. Write an if-else statement in PHP that checks whether a variable contains a string with more than 10 characters and prints a message accordingly.

A. # assume all necessary variables...

```
if (strlen($string) > 10) {  
    echo "This string has more than 10 characters.\n";  
}  
else {  
    echo "The string has 10 characters or less.\n";  
}
```

Instance of Operator

14. What does the instanceof operator do in PHP? Provide an example where it is used.

- A. The instanceof operator is used to check if an object belongs to a specified class or if it creates a particular interface.

Example:

#declare all necessary variables

```
<?php
```

```
class Car {}
```

```
class Lambo extends Car {}
```

```
$thisLambo = new Lambo();
```

```
if ($thisLambo instanceof Lambo) {
```

```
    echo "It would appear true that thisLambo is a Lamborghini.\n";
```

```
    if ($thisLambo instanceof Car) {
```

```
        echo "Yes, thisLambo is also classified as a car!\n";
```

```
    }
```

```
}
```

```
?>
```

showing how because the class is extended, the instance of lambo can also be checked.

There are other checks that can be used, including ones that would involve a statement that does not necessarily match the instance of class, but this provides a general idea of how it would function.

15. Why might you use instanceof in an object-oriented PHP application? Give a scenario where this would be beneficial.

- A. You may use instanceof in an object-oriented PHP application if you would like to verify or keep in check with a certain class type when you are writing your code. For instance, like the instance we used before, we classified lamborghini as a car. If we tried to use something else, like a house, thislambo would not be able to recognize a house as the same class as a car. It is scenarios like these where one might find an instanceof operator beneficial, and why it should be utilized.

16. What will the following code output, and why?

A.

```
class Animal {}
```

```
class Dog extends Animal {}
```

```
$dog = new Dog();
```

```
var_dump($dog instanceof Animal);
```

The following code will output bool(true) because once again we see that in the class of animal, a dog is recognized as the subclass, and in other words, mydog is indeed an (instanceof) animal.