Names: YUMVAGUSENGA Patrick

CLASS: ITA

REGNO:21RP05569

Q1. PHP programming is one of the most useful tools you can learn at the start of your Web Development career, as it's gives you access to plenty of great jobs, PHP is an acronym for: Hypertext Preprocessor" PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use.

Q2.

- **It's easy to learn and use:** One of the main reasons PHP became so commonplace is that it is relatively simple to get started with
- **It's open source (and therefore free!):** This also helps developers get started with PHP - it can be installed quickly and at zero cost
- **It's versatile:** One of the major benefits of PHP is that it is platform independent, meaning it can be used on Mac OS, Windows, Linux and supports most web browsers
- **It enjoys strong community support**: As a veteran scripting language that is widely used, PHP now has a large and loyal community base to support it
- **It's fast and secure:** Two things that every organization wants their website or application to be are fast and secure
- **It is well connected with databases:** PHP makes it easy to connect securely with almost any kind of database

Q3. the latest php version we have today is **8.2.0** and the updated features for the latest 3 are **8.0.26, 8.1.13**

Q4. **A new release: is** a change or set of changes that is created to be delivered to the production environment **while A stable release** is a version of a software package that has been tested and verified. It is the latest (and sometimes final version) of a program that is considered safe for public use. This kind of release is also called a "Stable" release.

Q5. **Simple:** It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world.

Interpreted: It is an interpreted language, i.e. there is no need for compilation.

Faster: It is faster than other scripting language e.g. asp and jsp.

Open Source: Open source means you no need to pay for use php, you can free download and use.

```
*/
```

Platform Independent: PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.

Case-sensitive: PHP is a partially case-sensitive language. Although functions names are not case-sensitive, other things in PHP are case-sensitiveQ6. Case Sensitive

Q6 .PHP classes are a mix between variables and functions, so they are partially case-sensitive. As you can see in the example above, the variables $num and $NUM can have different values. But when you declare two functions with the same name, PHP produces a fatal error: cannot redeclare the function

Q7. A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code. Examples

1. **This is a single-line comment**

```
<!DOCTYPE html>
<html>
<body>
<?php
// This is a single-line comment
# This is also a single-line comment
?>
</body>
</html>
2. <!DOCTYPE html>
<html>
<body>
```

2. **multiple-lines comment**

```
<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
```

```
?>
</body>
</html>
```

Q8.

A) **Echo:** In PHP, Echo acts as a statement that is used to show the output. It does not return any value and has the ability to pass multiple strings split by comma (,) in echo. We can use the Echo statement with and without parentheses, and it is faster in nature.

- **Print: In** PHP, the Print statement is also used to show the output. We can use it as an alternative to Echo. However, it is slower than Echo and returns an integer value 1. Also, in the Print statement we cannot pass multiple arguments.

1.	In Echo, we can pass multiple arguments separated by commas.	In Print, we cannot pass multiple arguments.
2.	In Echo, we can exhibit the outputs of one or more strings separated by commas.	Through the Print statement, we can only show the strings.
3	Echo can be used with or without parentheses	Print can also be used with or without parentheses.
4.	It never returns any value.	It always returns the integer value that is 1.
5.	This statement is fast as compared to the print statement.	It is slow as compared to the echo statement.

b) print vs printf

- printf() outputs a formatted string whereas print() outputs one or more strings.

```
        ex:     <?php
                print "Hello, world!";
                ?>
                output: Hello, world!
        ex:     <?php
                $number = 8;
$str = "Solar System";
printf("There are %u planets in the %s.", $number, $str);
?>
output:         There are 8 planets in the Solar System.
```

C) printf() vs print_r()

The printf( ) function builds a formatted string by inserting values into a template. The print_r( ) function is useful for debugging—it prints the contents of arrays, objects, and other things, in a more-or-less human-readable form.

```
                ex1:    <?php
                $number = 8;
$str = "Solar System";
```

```
printf("There are %u planets in the %s.", $number, $str);
?>
output:            There are 8 planets in the Solar System.
```

ex2:

```
class P {
  var $name = 'nat';
  // ...
}

$p = new P;
print_r($p);
```

```
output
Object

(

    [name] => nat

)
```

d) print_r() vs var_dump()

The var_dump() function displays structured information about variables/expressions including its type and value. Whereas The print_r() displays information about a variable in a way that's readable by humans.

```
$arr = array ('xyz', false, true, 99, array('50'));
```

output for print_r($arr)

```
Array
(
    [0] => xyz
    [1] =>
    [2] => 1
    [3] => 99
    [4] => Array
        (
            [0] => 50
        )
)
```

output for var_dump($arr)

```
array(5) {
  [0]=>
  string(3) "xyz"
  [1]=>
  bool(false)
  [2]=>
  bool(true)
  [3]=>
  int(100)
  [4]=>
  array(1) {
    [0]=>
    string(2) "50"
  }
}
```

Q9. **Data Types:** define the type of data a variable can store.

- **Integer:** Integers hold only whole numbers including positive and negative numbers, i.e., numbers without fractional part or decimal point
- **Double**: Can hold numbers containing fractional or decimal parts including positive and negative numbers or a number in exponential form
- **String:** Hold letters or any alphabets, even numbers are included. These are written within double quotes during declaration
- **Boolean:** Boolean data types are used in conditional testing. Hold only two values, either TRUE(1) or FALSE(0). Successful events will return true and unsuccessful events return false
- **Array:** Array is a compound data type that can store multiple values of the same data type. Below is an example of an array of integers

Q10**. A variable:** Is A memory Zone to store data. A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume)

Rules for PHP variables:

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

- Variable names are case-sensitive ($age and $AGE are two different variables)

Q11.

All of the superglobal variables act as associative arrays that use a string value as a key to access values. The following is a list of superglobal variables in PHP:

- **$GLOBALS** is the super global variable that stores all user-defined global variables. The global variable names act as keys to their values.
- **$_SERVER** contains data about headers, scripts, and paths. The keys to the values in this array are predefined.

- **$_REQUEST** stores data input in the form of HTTP POST, GET and Cookies. The keys to this array are defined in the HTTP requests.

- **$_POST** stores data input in the form of POST requests. The keys to this array are defined in the HTTP POST request.
- **$_GET** has data input in the form of GET requests. The keys to this array are defined in the HTTP GET request.

- **$_FILES** is a two-dimensional associative array that contains a list of files that were uploaded to the script using the POST method. The keys to this array are the names of the fields uploading the files and the data being accessed

- **$_COOKIES** keeps data input via HTTP Cookies. The keys to this array are defined when the cookies are set.
- **$_SESSION** holds session variables. Session variables can be accessed on multiple pages. This array's keys are defined by the users when they define session variables.

- **$_ENV** contains information about the environment that PHP is running in. The keys to the values in this array are predefined.

References

https://www.jobsity.com/blog/8-reasons-why-php-is-still-so-important-for-web-development https://www.php.net/releases/index.php

https://byjus.com/gate/difference-between-echo-and-print-in-php/

https://www.w3schools.com/