RWANDA POLYTECHNIC

IPRC TUMBA
Integrated Polytechnic Regional College

P.O. Box 6638 Rulindo
Tel: +250 786000011
Email:info@iprctumba.rp.ac.rw
www.iprctumba.rp.ac.rw

Name: IZERE Ange Elyse

RegNo: 21RP03300

L6Y2 A

Develop backend using PHP

ASSIGNMENT 2

Q1. The term PHP is an acronym for PHP(Hypertext Preprocessor). PHP is a server-side scripting language designed specifically for web development. It is open-source. It is very simple to learn and use. The files have the extension ".php".

Q2.we need PHP programming because PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails, etc. and also php programming it was used to create dynamic web pages, developers prefer to use this scripting language for building the server side of web applications.

Q3. the latest php version we have today is 8.2.0

1. Attributes: also known as annotations in other languages, provide a mechanism to add metadata to your codebase. Attributes can be used with classes, methods, functions and properties. They're easier to work with than the docblock approach adopted by many PHP 7 projects.
2. Named Arguments : enable you to pass parameters to methods and functions using an array-like construct. This makes it simpler to skip optional parameters and pass them out-of-order.
3. Constructor Property Promotion Populating a class's properties with initial values is one of the most frequent roles of a constructor.

Q4. new release vs stable release of a software product

stable release of a software is version of a software that has been tested and verified while stable release of a software product Is the distribution of the final version or the newest version of software development

Q5. Features of PHP programming

- Simplicity it means is organized and easy to learn. Even beginners won’t face any hard time learning and using PHP. It is a very well-organized programming language, and it comes with a lot of pre-defined functions, which makes the task of the programmer easy

- Flexibility PHP scripts can run on any device- mobile, tablet, or PC. It is very compatible with various databases. It can be easily embedded and integrated into HTML, XML, and JavaScript.

- Objective oriented PHP support data encapsulation, inheritance, abstraction, polymorphism, etc. The Object-oriented programming feature was added in PHP5. This feature helps in building complex reusable web pages and makes PHP comparable to powerful object-oriented languages like Java and Python.

- Interpreted language it means there is no need for compilation. Interpreters run through a program line by line and execute the code. Since interpreters execute source code themselves, the code becomes platform-independent.

- Efficient PHP is a versatile, reliable, and efficient programming language. The memory management of PHP is very efficient. Great session management, eliminating unnecessary memory allocation, are some of the features that make PHP efficient.

- Fast Performance PHP scrips are usually faster than other scripting languages. Users can load their web pages faster, and they love it. PHP code runs faster than most of programming languages because it runs in its own memory space. Not only that , but its connection with databases is also fast and efficient.

- Free and open-source it means it can be downloaded and used freely. There is absolutely no hassle to acquire a license to use it and no payment is required to use it, so it is kind to your pocket too!

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

  The following things in PHP are case-sensitive:

  - Variable names
  - Constructs (if, if-else, if-elseif, while, do-while)
  - Keywords (such as true and false)
  - User-defined functions and class names

- Security PHP has many pre-defined functions for data encryption. Users can also use third-party applications for security. Security and flexibility are often contrasting features, but PHP somehow manages to offer them both, and that’s great. PHP is designed specifically to be a more secure language for writing CGI (Computer-generated Imagery) programs.. Filter_var and strip_tags functions help to keep the environment more secure and safe for users.

- Platform independent We can run PHP on any device and operating system (Microsoft Windows, macOS, Linux, RISC OS, or Unix). We can easily connect it with various databases and is also compatible with almost all web servers used today (Apache, IIS, and others). It supports a wide range of databases as well. Its cross-platform compatibility makes really popular among its users as it saves a lot of time and energy.

- Loosely typed language PHP supports variable declaration without declaring its data type.

- Real-time access monitoring PHP provides real-time information about users' access. It provides a summary of recent accesses of the user. PHP offers a secure user management system and prevents unrestricted access.

- Error reporting and handling PHP has many pre-defined functions and reporting constants that generate errors at runtime.PHP5 allows you to use semantics like try, throw and catch, like Java and C#. There are 16 levels of error in PHP5, representing the category and severity of an error in PHP. Also, error reporting in PHP is super easy with function error_reporting().

- Memory and CPU use information PHP has functions like memory_get_usage() and memory_get_peak_usage(), which provide users with the memory usage information. To get the current memory usage, we can use the memory_get_usage() function, and to get the peak amount of memory used at a point, we can use the memory_get_peak_usage() function. Therefore, by using these functions, you can monitor your memory usage and optimize your code by eliminating the lines of code consuming a lot of memory resources.

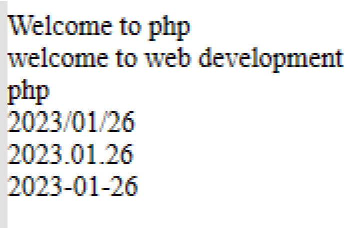
Q6. PHP Case Sensitivity

PHP functions, classes, core language keywords are case-insensitive. That means you can write classes, functions and keyword in any case.

For example, you can write echo construct in different ways like Echo, ECHO, echo or you can write date function like Date, DaTe, date.

```
<?php

echo "Welcome to PHP". "<br>";

ECHO "welcome ". "<br>";

Echo "php". "<br>";

echo Date("Y/m/d") . "<br>";

echo DATE("Y.m.d") . "<br>";

echo Date("Y-m-d");

?>
```

<u>out put</u>



```
Welcome to php
welcome to web development
php
2023/01/26
2023.01.26
2023-01-26
```

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.

Example:

```
<!DOCTYPE html>

<html>

<body>

<?php

// This is a single-line comment
```

```
# This is also a single-line comment
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>
</body>
</html>
```

Q8. Differentiate with real example the following php output functions:

a.Echo() vs print(): echo and print are more or less the same. They are both used to output data to the screen.

PRINT

```
<?php
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

OUTPUT

```
Hello world!
I'm about to learn PHP!
```

ECHO

```
<?php
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";

?>
```

OUT PUT

```
Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.
```

b. Print() vs printf(): printf () outputs a formatted string whereas print () outputs one or more strings.

PRINTF()

```
<?php

   $val = 2976;

   printf("%f",$val);

?>
```

Output

```
2976.000000
```

PRINT()

```
<?php

print "Hello world!<br>";

print "I'm about to learn PHP!";

?>
```

OUTPUT

```
Hello world!
I'm about to learn PHP!
```

c. Printf() vs print_r()

PRINTF()

```
<?php
```

```
   $val = 2976;

   printf("%f",$val);

?>
```

Output

```
2976.000000
```

PRINT_R()

```
<?php

$a = array("red", "green", "blue");

print_r($a);

echo "<br>";

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

print_r($b);

?>
```

Output

```
Array ( [0] => red [1] => green [2] => blue )
Array ( [Peter] => 35 [Ben] => 37 [Joe] => 43 )
```

d. Print_r vs var_dump()

PRINT_R()

```
<?php

$a = array("red", "green", "blue");

print_r($a);

echo "<br>";

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

print_r($b);

?>
```

Output

```
Array ( [0] => red [1] => green [2] => blue )
Array ( [Peter] => 35 [Ben] => 37 [Joe] => 43 )
```

var_dump()

```
<?php

$a = 32;

echo var_dump($a) . "<br>";


$b = "Hello world!";

echo var_dump($b) . "<br>";

?>
```

OUTPUT

```
int(32)

string(12) "Hello world!"
```

Q9.Data Types generally represent the type of value that we can store in a Variable. PHP Data Types are majorly classified into three groups which are Scalar, Compound, and Special.

Scalar Data Types in PHP

Scalar data types are used to store single values, and these are the most widely used data types in PHP. There are four types available in this category, and please find below the details about them.

Boolean data type used to store true or false values in a variable. These boolean values are default keywords, and they should not be enclosed with single or double quotes.

Integer data type is used to store numbers that can be positive, negative, hexadecimal, and octal values. Integer values should not have spaces, commas between them, and by default, it is considered a positive value

Float/Double data types are considered Floating point numbers or Doubles or Real numbers, which are decimal numbers with fractional point values. It can be positive or negative values

String data type can hold a sequence of characters which can be anything, including strings, numbers, or special characters. PHP String has to be enclosed with single quotes or double quotes with necessary escape sequence characters.

Compound Data Types in PHP

Compound data types are used to store more than one value in a variable. The values can be the same data type or different data types: numbers, strings, etc.

Array data type is used to store multiple or series of values which can be any type. The array values are associated with their key, so usually, the key is used to get their values.

Objects is a type of data that allows storing data and obtaining information about how that data is processed. PHP Objects specifically created for Classes in PHP, and it is an instance for that Class to access their functions and properties

Callable/Callback type declarations can represent the PHP Callbacks function. PHP user-defined functions are used as parameters in built-in functions like call_user_func, and this function invoked the specified user-defined function for processing.

Iterable data type is used to set a specific parameter as an iterable variable, indicating that the function requires a set of values for that parameter. If that specific parameter’s value not traversable, then it will throw an error. Please find below the example,

Special Data Types in PHP

Special Data Types holds a value that is not a common case but special. Please find below the details about them.

NULL data type is used to store NULL values. It will not store anything else other than NULL. This kind of variable’s values are case-insensitive that we can assign NULL or null in the program.

Resource data type variables holds the value which provide reference information to the external resources like Database, Files, etc.,

Q10. Variables are the identifier of the memory location, which used to save data temporarily for later use in the program.The purpose of variables is to store data in memory for later use.

the variable naming rules you have to obey while defining a variable in php

- 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. The PHP super global variables are:

- $GLOBALS : is used to access global variables from anywhere in the PHP script (also from within functions or methods).
- $_SERVER : is a PHP super global variable which holds information about headers, paths, and script locations.
- $_REQUEST : is used to collect data after submitting an HTML form.
- $_POST : is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.
- $_GET : is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".
- $_FILES : is a variable in PHP that is used to store objects uploaded using the HTTP POST method.

- $_ENV :is super global associative array in PHP. It stores environment variables available to current script.
- $_COOKIE: is a tiny file that a server places on a user's computer. It establishes the user's identity. Whenever a server request is made.
- $_SESSION: is a PHP super global that keeps and provides information about a site user from the moment the site is opened until it is closed.

References

PHP | Introduction - GeeksforGeeks

Top Features of PHP You Must Know - InterviewBit

PHP 5 Case Sensitivity - pTutorial

PHP Echo and Print Statements (w3schools.com)

Data Types in PHP 7 - WritePHPOnline

PHP $_GET (w3schools.com)