

PHP - Assignment - 2

(1) Explain & define the class and objects with example.

↳

1) CLASS :-

=

→ classes are the blueprints of objects. One of the big difference between functions and classes is that a class contains both data variables and functions that form a package called an 'object'.

→ class is a programmer-defined data type, which includes local methods and local variables.

→ class is a collection of objects.

Syntax :

=

```
<?php
class className {
    //data members
    //member functions
}
```

Example :

=

```
<?php
class Person {
    public function sayHello() {
        echo "Hello </b>";
    }
}
```

objects:

- In PHP, Object is a compound data type (along with arrays).
- Values of one types can be stored together in a single variable.
- Object is an instance of either a built-in or user-defined class.
- In addition to properties, class defines functionality associated with data.

Syntax:

```
$obj = new className;
```

Example: (e.g. printing) = 220

```
<?php  
class SayHello {  
    function hello () {  
        echo "Hello World";  
    }  
}
```

```
$obj = new SayHello;  
$obj->hello();  
?>
```

(2) How to read a file? Explain its file handling functions.



→ File handling is an important part of my web application. You often need to open and process a file for different tasks.

→ To read file PHP provides `readfile()` function that reads a file and writes it to the output buffer.

→ Assume we have a text file called "webdictionary.txt", stored on the server, that looks like this:

AJAX = Asynchronous Javascript and XML
CSS = Cascading Style Sheets
HTML = Hypertext Markup Language

Example:

```
<?php  
echo readfile("webdictionary.txt");  
?>
```

File handling Functions:

(1) `fopen()`

→ The PHP `fopen()` function is used to open a file.

Syntax:

resource fopen (string \$filename, string \$mode)

[, bool \$use_include_path=false], resource
\$context])

example: // writing file

<?php

\$handle = fopen("c://folder//file.txt", "r");

(2) fclose()

→ The PHP fclose() function is used to close an open file pointer.

Syntax:

bool fclose(resource \$handle);

example:

<?php

{}

(3) fwrite()

→ The PHP fwrite function is used to write content of the string into file.

Syntax:

int fwrite(resource \$handle, string \$string [, int \$len])

example:

<?php

\$fp = fopen('dutu.txt', 'w');

fwrite(\$fp, 'Hello');

fwrite(\$fp, 'PHP file');

fclose(\$fp);

}

(3) Explain constructor in brief.



→ A constructor allows you to initialize an object's properties upon creation of the object.

→ If you don't create an object constructor function, PHP will automatically call this function when you create an object from a class.

→ constructor function starts with two underscore (-).

Syntax: <?php

```
class className {  
    function __construct() {  
        echo 'constructor is called';  
    }  
}
```

example:

<?php

```
class Fruit {  
    public $name;
```

function __construct(\$name) {

\$this->name = \$name;

}

?php

}

- (4) Define inheritance and abstract class.
- Inheritance in OOP = When a class derives from another class.
 - The child class will inherit all the public and protected properties and methods from the parent class.
 - An inherited class is defined by using the extends keyword.

Syntax:

```
<?php
```

```
class className{
```

```
    // properties
```

```
}
```

```
class derivedClassName extends mainClassName {
```

```
    // properties
```

```
}
```

```
}
```

example:

```
<?php
```

```
class fruit {
```

```
    public $name;
```

```
    public $color;
```

```
}
```

```
class strawberry extends fruit {
```

```
    public function message() {
```

```
        echo "I am strawberry";
```

```
}
```

```
}
```

abstract class :

- Abstract classes and methods are when the parent class has a named method, but needs its child class to fill out the tasks.
- An abstract class is a class that contains at least one abstract method.
- An abstract class or method is defined with the 'abstract' keyword.

Syntax: <?php

abstract class ParentClass {

 abstract public function SomeMethod1();

 abstract public function SomeMethod2();

 (\$name, \$color);

}

}

(5) Explain \$ cookies and \$ session with example.

cookies:

→ A cookie is a small piece of data that's stored in user's web browser.

→ It can be used to store information such as user's preferences or login information.

→ When the user visits a website, the server can send a cookie to the user's browser, which the browser will then store.

→ To use cookies in PHP, you can use the setcookie() function.

Syntax:

```
setcookie('name of cookie', value, exp. time);
```

example:

```
<?php
```

```
setcookie('fav-color', 'blue', time() + (86400 * 30));
```

```
?>
```

Session:

→ A session is a way to store information to be used across multiple pages.

→ When a user visits a website and starts a new session, the server creates a

unique session ID and stores it in a cookie on the user's computer.

→ Sessions are useful for storing temporary data that is specific to a single user and a single browser session.

→ For example, you might use a session to store a user's shopping cart items or login status.

→ To use session in PHP, you first need to start a session using the `session_start()` function.

Ex:

```
<?php  
Session_start();  
$_SESSION['fav-col'] = 'blue';
```

(6) Explain PHP OOPS concept with example.

→ In PHP, you can also write PHP code in an object-oriented style.

→ Object oriented programming is faster and easier to execute.

→ OOP stands for Object-Oriented Programming.

→ Classes and Objects:

→ Classes and objects are the two main aspects of object-oriented programming.

→ Class is a template for objects, and an instance is object is an instance of a class.

Define a class:

```
<?php  
class Fruit {  
    // code goes here  
}
```

Define object:

```
<?php  
$obj = new Fruit();  
?
```

The \$this keyword:

→ The `$this` keyword refers to the current object, and is only available inside methods.

example:

`<?php`

```
class Fruit {
    public $name;
    function set_name($name) {
        $this->name = name;
    }
}
```

`?>`

Inheritance:

→ Inheritance in OOP = when a class derives from another class.

→ the child class will inherit all the public and protected properties and methods from the parent class.

Ex.

`<?php`

```
class Fruit {
```

```
    public $name;
```

```
    public $color;
```

`}`

```
class strawberry extends Fruit {
```

```
    public function message() {
```

```
        echo "Hello";
```

`}`

`}`

(3) what is exception ? Explain error handling try and catch block with example.

4

→ Exceptions are used to change the normal flow of a script if a specified error occurs.

→ Exception handling is a powerful mechanism of PHP, which is used to handle runtime errors, so that the normal flow of the application can be maintained.

→ The main purpose of using exception handling is to maintain the normal execution of the application.

→ Exception handling in PHP is almost similar to exception handling in all other programming languages.

→ By using exception handling we can separate error handling code from normal code.

→ We can also group of error types.

→ An exception is an unexpected outcome of a program, which can be handled by the program itself.

try - catch

→ when a PHP exception is thrown, the PHP runtime system looks for a corresponding 'catch' statement that can handle specific exception type.

Syntax:

```
<?php
```

```
try {
```

(run your code here)

```
}
```

```
catch (Exception $e) {
```

(code to handle exception)

```
}
```

→ The try block contains the code that may potentially throw an exception.

Ex: <?php

```
function checkNum ($number) {
```

```
if ($number > 1) {
```

throw new Exception ("Value must be
1 or below");

```
return true;
```

```
try {
```

```
checkNum(2);
```

echo "If you see this, the number is below";

```
catch (Exception $e) {
```

echo 'message: ' . \$e->getMessage();

```
}
```

```
?>
```

(8) What is Joomla?

Joomla is an open-source content management system (CMS) that enables users to build websites and powerful online applications. It's written in PHP and uses a MySQL database to store content.

Joomla is known for its flexibility, ease of use, and extensive community support, making it popular choice for creating various types of websites.

(9) Explain Advantages and disadvantages of Joomla.

Advantages:

(1) User-Friendly Interface: Joomla offers an intuitive and user-friendly interface, making it easy for users to manage and update their websites.

(2) Flexibility: Joomla provides a high level of flexibility, allowing users to customize and extend their websites.

(3) Community Support: Joomla has a large and active community of developers and users.

(4) Security: Joomla takes security seriously and regularly releases updates.

Disadvantages:

(1) Learning curve: While Joomla is user-friendly, it still has a learning curve, especially for beginners.

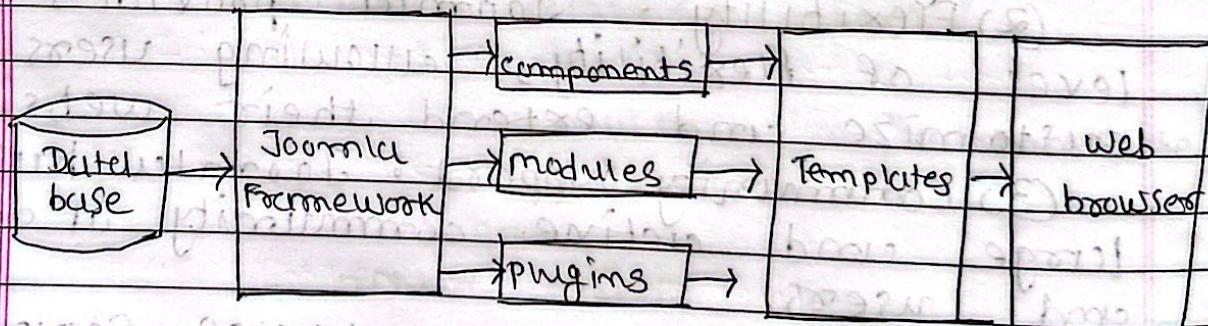
(2) Performance: Joomla can be resource-intensive, especially when using a large number of extensions and plugins.

(3) Customization complexity: Customizing Joomla beyond its default setting may require some technical expertise.

(10) Explain Joomla Architecture.

→ Joomla is a Model-View-Controller web application.

→ The following diagram shows the architecture of Joomla.



→ The architecture of Joomla contains following layers:-

- Database
- Joomla Framework
- components
- modules
- plugins
- Templates
- web server

• Database :

→ Database is a collection of data and can be stored, manipulated and organized in a particular manner.

• Joomla Framework :

→ Framework is a collection of open source software, where the Joomla CMS is built.

• Components :-

→ Components are considered as mini applications. It consists of two parts i.e. Administrator and site. Whenever a page gets loaded, component is been called to render the body of main page.

• Modules :

→ Modules is .cm extension which is used to render the pages in Joomla. It is also used to display the new data from the component. It frequently looks like boxes such as login modules.

• Plugins :-

→ This is also a kind of Joomla extension, it is very flexible and powerful for extending the framework. It contains a bit of codes that is used to execute the particular event trigger.

- **Templates :**

→ Templates determines the look of the Joomla website. There are two types of templates used i.e. Front-end and Back-end. The backend template is used to control the functions by the administrator.

- **Web-browsers :**

→ It is a server where the user interacts. It delivers the web pages to the client. The HTTP is used to communicate between the client and the server.

(11) Write steps to create menu in Joomla.



Step 1: Log in to the Joomla Administrator Dashboard.

Step 2: Navigate to the menu manager.

Step 3: Create a new menu.

Step 4: Enter menu details.

Step 5: Save Your Menu.

Step 6: Add menu Item's.

Step 7: Configure menu Item

step 8 : Save Your menu Item.

step 9 : Repeat for Additional menu Items.

Step 10 : Assign menu module.

(12) How to create and add new article
in Joomla. write down steps.



step 1 : Log in the Joomla Administrator Dashboard.

step 2 : Navigate to the Article Manager.

step 3 : Create a New Article.

step 4 : Enter your article details.

step 5 : Add content to Your Article.

step 6 : Configure Article options.

step 7 : Once you have entered all the details save your Article.

step 8 : View your Article.

