Introduction to OOPs

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In Earlier days, in order to write the bigger bigger logic, we used two Approaches to write the code

1. mono-lithic: - when we write the code, in single script itself.

2. poly-lithic: - When we write the entire code, as a group of small small scripts then it is called as poly-lithic approach.

a. Using functional Approach

1. Using Object Oriented Approach
2. Using modular Approach
3. And other Approach

These All approaches for writing the code, is called as, Programming Paradigm or Methodologies, or Discipline, or Technique, Programming Mechanism.

**What is Object Oriented Programming System?**

1. It is a Programming Paradigm or Mechanism for developing Software Highly Scalable Application.

**Object Oriented Programming Language: -**

Any programming Language which can provide OOPs support with all features implemented by OOPs that language is called as OOPL. (Object Oriented Programming Language)

Eg: - C++, Java, C#, PHP, Python….etc

Pillars of Object-Oriented Programming Language

-----------------------------------------------------------------------

1. Encapsulation

2. Data Hiding

3. Data Abstraction

4. Abstraction

5. Inheritance

6. has-Relationship

7. is-Relationship

8. Class and Objects

9. Polymorphism

10. Dependency Injection

11. Exception Handling

After the Object-Oriented Programming System has left its scope: it leaves some elements as a Scrap. These elements are then re-used and converted into re-usable components are These re-usable components of, OOPs are called as Design Patterns.

There is total 23 design patterns

1. Structural

Decorator

Bridge

Adapter

Flyweight

Facade

Composite

Proxy

1. Behavioural

3. Creational

These are written by 4 Writers and called as Gangs of 4.

Advantage of OOPs: - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. easy to Understand

2. Bottom to Top Approach

3. Security

4. Reusability

5. Scope and Memory Management

6. Standardisation

Q. Is php a Object Oriented Language?

Ans. Yes Over-view, Detail No

Java ---> Yes Over-View, Detail No

C# ---> Yes Over-View, Detail No

Python ---> Yes Over-View, Detail No

C++ ---> Yes Over-View, Detail No

Smalltalk ---> Yes

Any Programming is said to be fully OOPs based if it does not have primitive datatype. And if it has then it should be on the basis of, Class and Object.

How to make class in and object in PHP

----

Important Points Related to Constructor: - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. it is called as Instantiation of the class i.e., as soon as Object is created.

2. In case of PHP, constructor are like methods, with no return types.

Q. method and constructor

Ans: i. Method can return anything : getter.

ii. constructor method cannot return anything : constructor.

3. Constructor acts as a setter which can be used to initialise the data member, and assign null to the unused variable.

Constructor uses garbage collector mechanism for un-used variable.

---------------------------------------------------------------------------------------------------------

**Context in OOPs: -**

Meaning of context in OOPs is scope of Members and Member Functions with reference to memory, either Stack or either Heap.

What is Static?

static in php is concept

1. static scope in variable

2. static context in OOPs

NOTE :: static is defined in terms of logical entity not physical. As soon as we define a class and object, we have two context in that class.

1. Class Context(Scope) :: class Test

2. Object Context(Scope) :: $text = new Test;

Once you are able to analyse the context, it is easy to play with data member and member function on the basis of context.

Static context is more to class Then object… ----> use class name

How to use \*\*\*\*\*\*\*\*\*\*\*\*\*

Class name :: $<property>

Class name :: <function-name()>

Non-context is more to object than class.

---> use object name outside class.

---> use $this Inside the class.

How to use \*\*\*\*\*\*\*\*\*\*\*

$this->property

$this->function-name()

Case Study: -

<?php

//wap in php to show static context

Class Test{

public $a=300; #non-static context

public static $b=400; #static context

public function get\_a(){

echo “The value of a in static context is = {$this->a}”;

echo “The value from Non context = ”;

//echo Test::$a; #error due to accessing non-static a from static context

echo PHP\_EOL;

}

public static function get\_b(){

//echo “The value from Non static context {$this->a} \n”;

echo PHP\_EOL;

echo “The value from static context = ”;

echo Test::$b;

//echo $this->b; //error due to $this Object Context for static context b

}

}

$test = new Test();

$test->get\_a();

//Test::get\_a(); #error get\_a scope non static it cannot be called by class //context

$test->get\_b();

Test::get\_b();

**Inheritance: -** deriving the properties(member) and features(methods) from parent to child class is called as inheritance.

Inheritance may be defined as the mechanism of re-usability. Using the Readymade Items (Data Member + Member Function) from existing class is a concept of Re-usability in Inheritance.

**Basic terms Related to, Inheritance**

1. **multi-level class**

2. **Independent class:** if one class do not depend on any other class then it is called independent class.

3**. Dependent class (Relationship):** if any class utilises the, features of any existing ready-made class then this is called as dependent class and it is set-up a relationship b/w both of them.

**a. Has Relationship**: It is logical Relationship b/w one or more class. This is by-default Relationship and called Inheritance.

In order to achieve inheritance we need extends keyword.

<Child-class> will all Properties [extends] from <Parent>

In order to achieve inheritance we required at-least two multi-level classes.

i. parent class: This is independent class.

ii. child class: This is dependent class over parent class.

Syntax:

Class child extends Parent {

}

NOTE:: Parent class/Master class/ Base class/ Super Class/ Derived Class/ Baap Class/Inherited Class

Child Class/sub-class/inheriting class/deriving class/beta class

**b. Is Relationship:** It is Physical Relationship b/w one or more class. This is explicit Relationship and called as composition/aggregation.

In order to achieve **is relationship**, we need to explicitly make object of any class which you want to use.

Inside own case: -

Class A{

Public function B(){

$b = new B();

$b->someMethod();

}

}

Q. How will you sort a Array

Ans. Using pre-defined Function

1. sort() : sorting on values : ascending

2. rsort() : sorting on values in reverse order : descending order

1. assort : associative sorting on values
2. ksort : sort the keys in Ascending order.

5. arsort : reverse sorting in ascending order

6. krsort : reverse sorting on keys of associative array

**Types of Inheritance: - --------------------------------**

1. single Inheritance

2. multi-level Inheritance

3. Heiriechal Inheritance

4. multiple Inheritance

5. Hybrid Inheritance

6. Cyclic Inheritance

One of the Best use of Inheritance is Applied on file inclusion and In Laravel This concept is called as Template Inheritance.

What is file-inclusion?

Extending the source code from one file to another is called as file-inclusion.

1. include :

2. include\_once :

3. require :

4. require\_once :

**Types of error in PHP -------------------------------**

Note:: We have 16 types of Error level

Don’t get confused b/w Error level and Legal Error

Legal Error, will disturb the execution of the program.

Where as Error level will be raised only after program is successfully running.

1. **Parse Error :** if there Error occur, program execution will not take place.

2. **Syntax Error :** if these error occur, program execution will not take place.

But in case of Error level for e.g.: undefined index Error is Notice\_Error will generates Error, But do not the execution of the Program.

E.g. 2: Deprecation Error, Due to PHP Standards there may be a chance, that you are using the Latest Version of PHP but you are using some features which are of old legacy version in that scenario. You will get Deprecation Error instead of Terminating the program.

1. Syntax Error (Parse Error, Compilation Error):

If Occurred No single Line will be Execute.

2. Notice or Warning Error

If occurred then Notice error or Warning Error is raised. But, it will not affect the code just below the, Error line i.e. it do-not stop the, execution of the program.

3. Fatal Error:

If occurred then No line below the code will execute Ever until resolved.

4. PHP Standard Error or Exception:

This type of error, occur due to logical mistakes, which need to be either resolved or handled using Exception Handling, sometimes These Error can be critical like Fatal Error, Uncaught Error.

**Access Modifier -------------------------**

Access Modifier are verbs when added Before any Instance Variable or method, makes it hidden from the different scopes.

Possible Number of scopes:-

1. From Outside the class

2. From Inside the class

3. From Child class

Difference b/w Public, Private, Protected

location : Public | Protected | Private ------------------------------------------------------------------------------------------------------------------

Outside Class Valid | ---------- | ------------

Inside Class Valid | Valid | valid

Child Class Valid | Valid | ------------

**Method Chaining**

Chaining one method with different method without destroying the instance is called as method chaining.

It is widely used by Laravel, codeIgnitor, and MVC Framework