Apply OOPs in PHP

Created	@May 6, 2022 6:33 PM
Class	WBP
Type	Notes
Author	Swarup Vishwas

Classes

> Class is unit of code that describes the characteristics and behaviours of something. Which is blueprint or template of an object

Objects

> An *object* is specific instance of class, which can be made many times.

Properties

> The characteristics of a class or object are known as properties, and are nothing but variable stored

> in class

Methods

> Represents behaviour of class, the action associated in class are known as methods. Methods are very similar to Functions

Creating Class

1. Use `class` keyword
2. Followed by class name
3. Followed By paranthesis with function body
4. Syntax:
    a. `class Car{  }`

Creating Object

1. Declare normal variable(object)
2. use new keyword
3. Instantiate using class name
4. Syntax:
    a. `$obj = new Car();`

this Keyword

- The `$this` keyword allows us to approach the class properties and method from within the class
- It is psuedo-variable that references to current object

Differences between self and this keyword

self	this
Not preceded by any symbol	Preceded with `$` symbol
We use scope-resolution `::` operator here	We us `->` operator here
It is used to refer static members of the class	It is used to access non-static keyword
For static-member and not objects	**For current objects**
Eg	Eg
`self::<class_member>`	`this-><class_member>`

Constructor

> Constructor are special member function for initialize variable on the newly created object instances from a class. An object’s constructor method is called just after the object is created.

```
<?php
class MyClass{
  function __construct(){
    echo "Hello World..!!";
  }
}
```

```
$obj = new MyClass();
?>
```

Types of Constructor

1. Default
    a. It has no parameters, but the values to the default constructor can be passed dynamically
2. Parameterized
    a. It takes the parameters, and also you can pass different values to data members
3. Copy Constructor
    a. It accepts the address of the other objects as a parameter

Advantages of Using Constructor

- Constructor provides the ability to pass parameters which are helpful in automatic initialization of the member variable during creation time
- Can have many arguments and also default argument
- They encourage re-usability avoiding re-initializing whenever instance of class is created
- You can start session in constructor method so that you don’t have to start in all the function everytime
- They ca call clas member methods and functions
- They can call class member methods and functions
- They can call other Constructor even from Parent class

Destructor

> Is also a special member function which is exactly the reverse of constructor method and is called when an instace of class is deleted from the memory

> We create desctructor by using `__destruct` function

Constructor vs Destructor

Constructor	Desctructor
Accepts one or more arguments	No arguments are passed
__construct()	__destruct()
Same as class name	Class Name preceded with Tilda
When object is created	When object is destroyed
Used to initialize the instance of class	Used to de-initialize the instance of class
Can be overloaded	Cannot overload
Allocate memory	Deallocate memory
Multiple constructor allowed	Only one destructor allowed

Inheritance

> It is a concept of accessing the feature of one class from another class. If we inherit the class feature into another class, we can extends the features of a class by using “extends” keyword

- Types
  - Single
  - Multiple Inheritance (Interface)
  - Multi-level Inheritance

Single Inheritance

- Single inheritance is the concept in PHP in which one class can be inherited by a single class by
- Need two class

Multilevel Inheritance

- PHP supports Multilevel Inheritance
- In this types of inheritance, we will have more than 2 class
- In this types of inheritance, a parent class will be inherited by a child class then grandchild via child

Heirarchical Inheritance

- PHP supports Heirarchical Inheritance
- It is the type of inheritance in which a program consists of single parent and more than one child class

Traits

- Traits are used to declare methods that can be used in multiple classes
- Traits can have methods and abstract methods that can be used in multiple classes, and the methods can have any access

modifier

```
<?php
// For Define Traits
trait traitName{
  // some code
} ?>

<?php
// For Accessing Trait
class MyClass{
   use traitName;
}
```

Interface

> Interface allow you to specify what method a class should implement. To implement an interface, a class must use the `implements` keyword

Concrete Class

> The class which implements an interface is called the Concrete class

- It must implement all the method defined in an interface
- Interface of same name can’t be implemented because of ambuigity error

Abstract Class and methods

> Abstract class contains abstract methods and it just have name and args and no other code

Interface and Abstract class

Interface	Abstract
Interface class supports multiple inheritance feature	Abstract class does not support mulitple inheritance
This does not contain a data member	Abstract class does contain a data member
The interface does not allow container	Abstract class supports container
An interface class only contains incomplete members which refer to the sign of members	Abstract class can contain access modifiers within subs, functions and properties
Since everything is assumed to be public, an interface does not have access modifier by default	An abstract class can contain access modifiers within subs, functions and properties
Any member of an interface cannot be static	Only a complete member of abstract class can be static

Method Overloading

- Function overloading is the OOPs feature in PHP
- In function overloading, more than one function can have same method signature but different number of arguments
- Function overloading contains same function will have same method signature and number of arguments